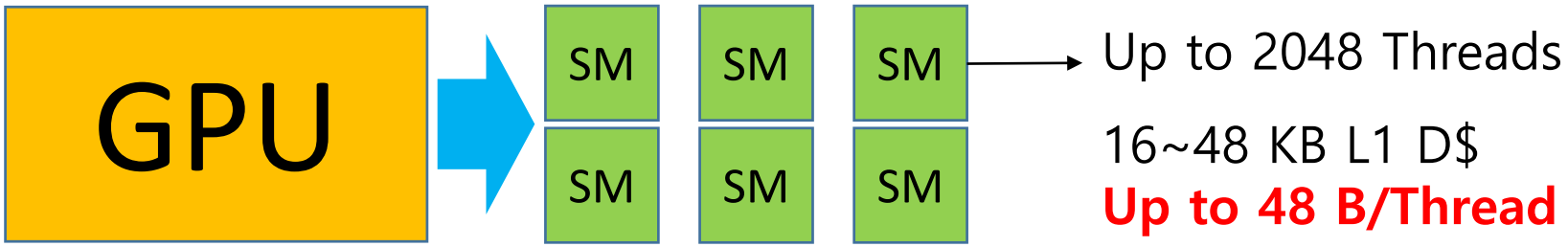
APRES:
Improving Cache Efficiency by Exploiting Load Characteristics on GPUs

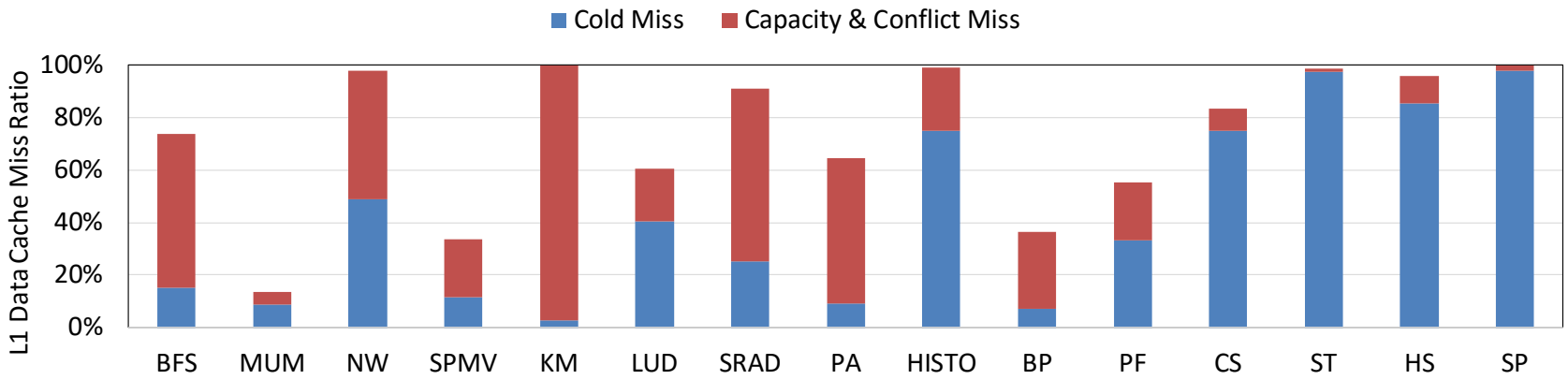
Yunho Oh, Keunsoo Kim, Myung Kuk Yoon,
Jong Hyun Park, Won Woo Ro (Yonsei University)
Yongjun Park (Hongik University)
Murali Annavaram (USC)

Inefficient Cache Utilization on GPUs

- **Small-Sized L1 Data Cache per Streaming Multiprocessor (SM)**

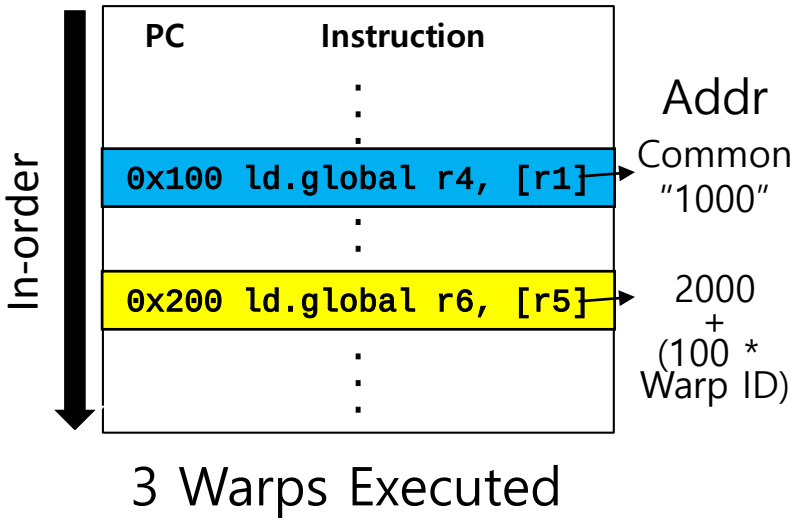


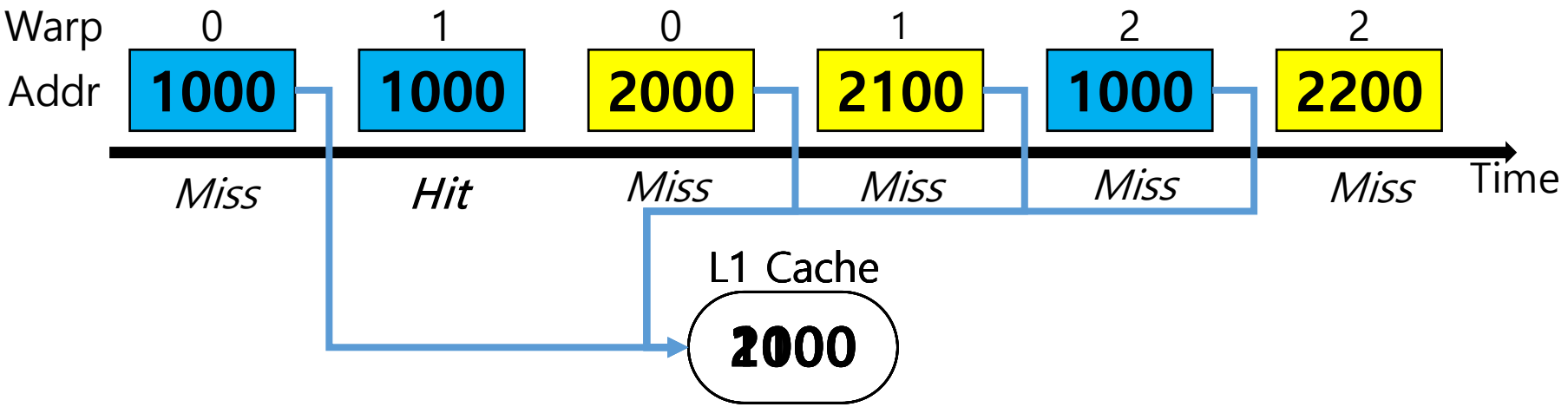
- **A Large Portion of Capacity/Conflict Misses (2Cs) on GPUs**
  - Decreases GPU performance due to Stalls in Memory Pipeline



Analysis of 2Cs

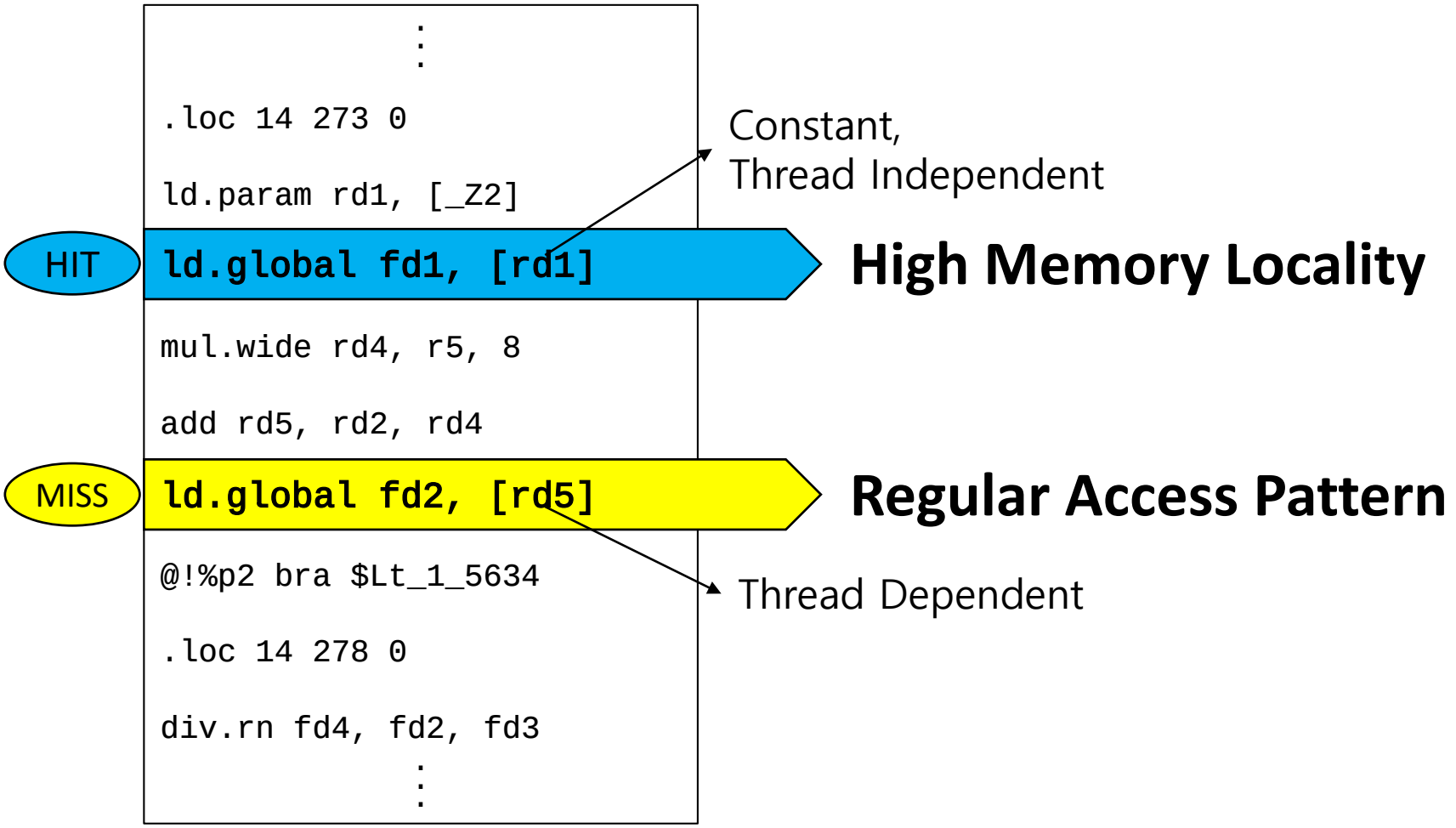
- **Warp 2 Accesses Address 1000.**
- **Address 1000 is Already Replaced.**





Unordered Load Executions Across Warps Increase 2Cs!

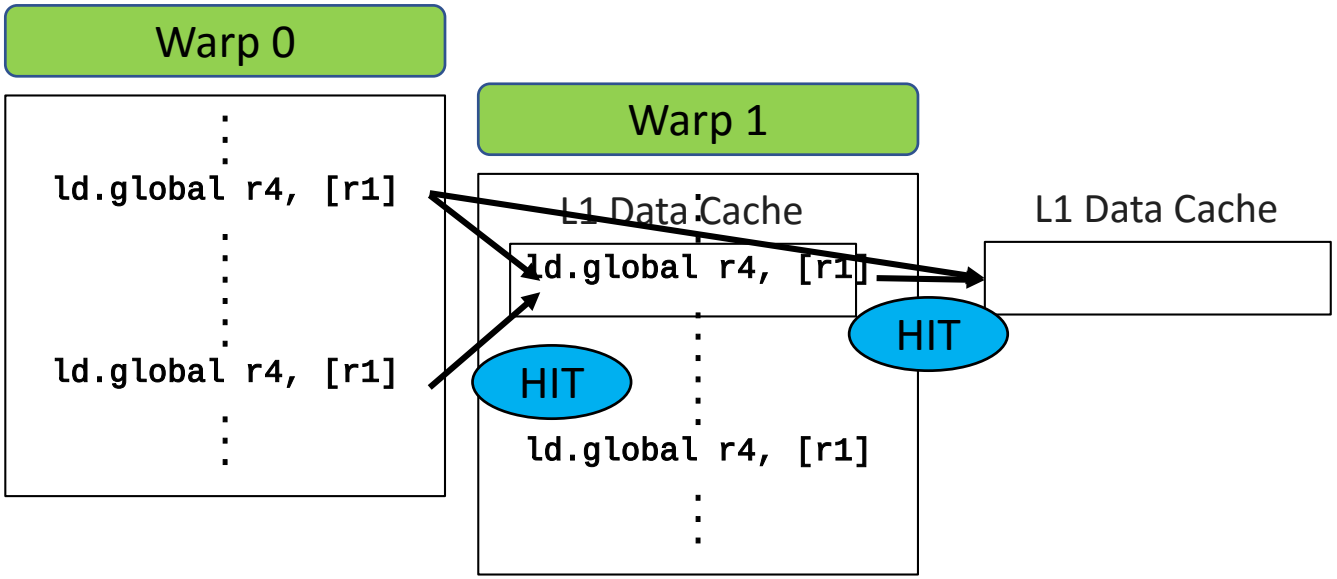
Classifying GPU Load Characteristics



Each Static Load Exhibits Same Cache Behavior "Across Warps".

Type 1: Loads Having High Locality

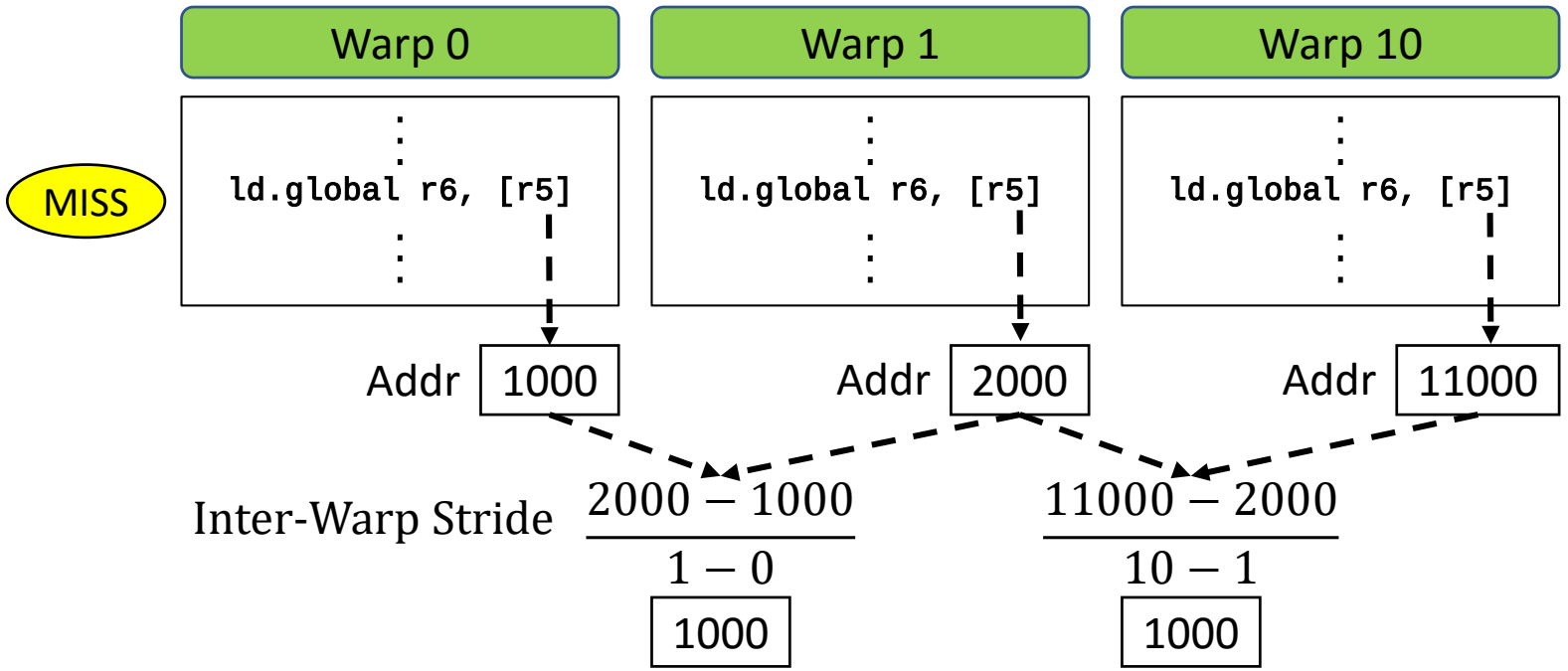
- **A Cache Line Can be Re-Referenced by**
  - Same Warp (Intra-Warp Locality)
  - Other Warps (Inter-Warp Locality)



Type 2: Loads Having Inter-Warp Strides

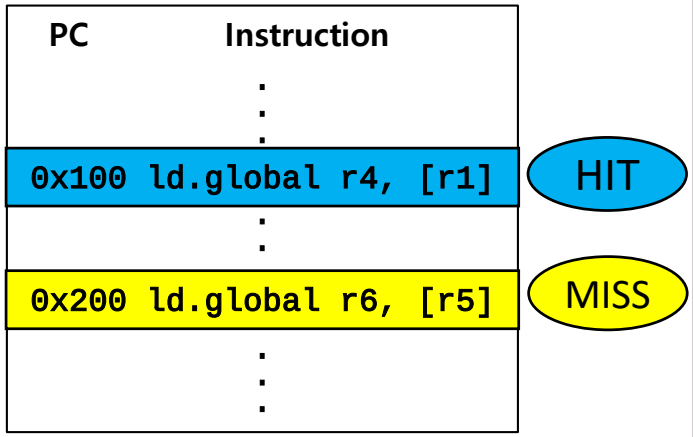
- **Regular Inter-Warp Strides**
- **Easy to Predict Memory Addresses to be Accessed In the Future**

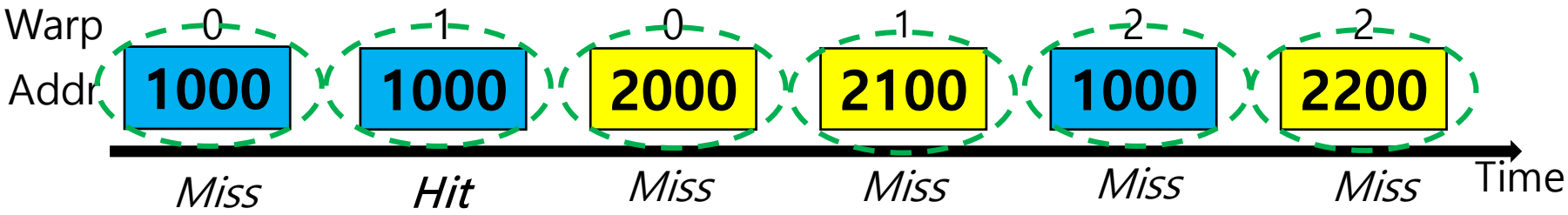
$$\text{Inter-Warp Stride} = \frac{\text{Addr2} - \text{Addr1}}{\text{Warp\_ID2} - \text{Warp\_ID1}}$$



Back to Example

- **Type 1 Loads Needs to be Executed Before Cache Lines are Evicted.**
- **Data Prefetching Increases Cache Hits in Type 2 Loads.**



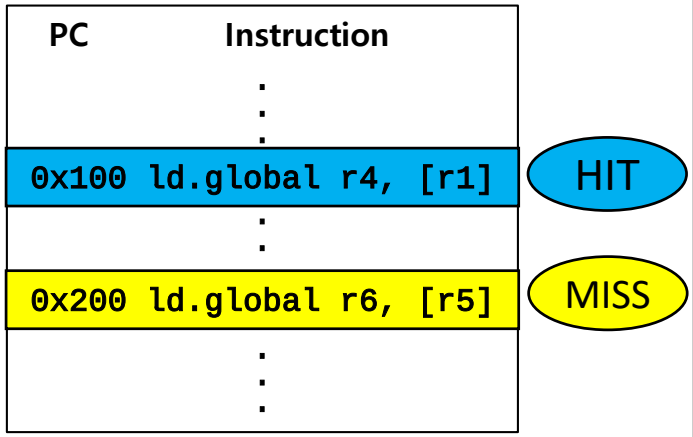


SOLUTION:

Adaptive PREfetching and Scheduling

How APRES Works

- **Warp Grouping and Prioritization**
- **Prefetching for Warp Issuing Same Static Load**



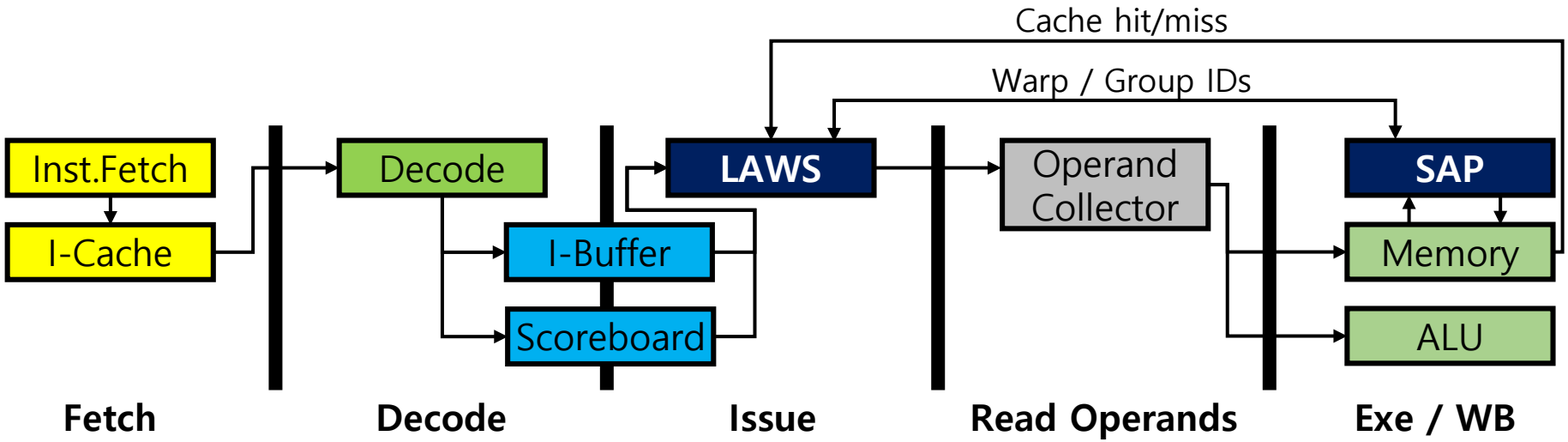
Previous Example

APRES

APRES Architecture

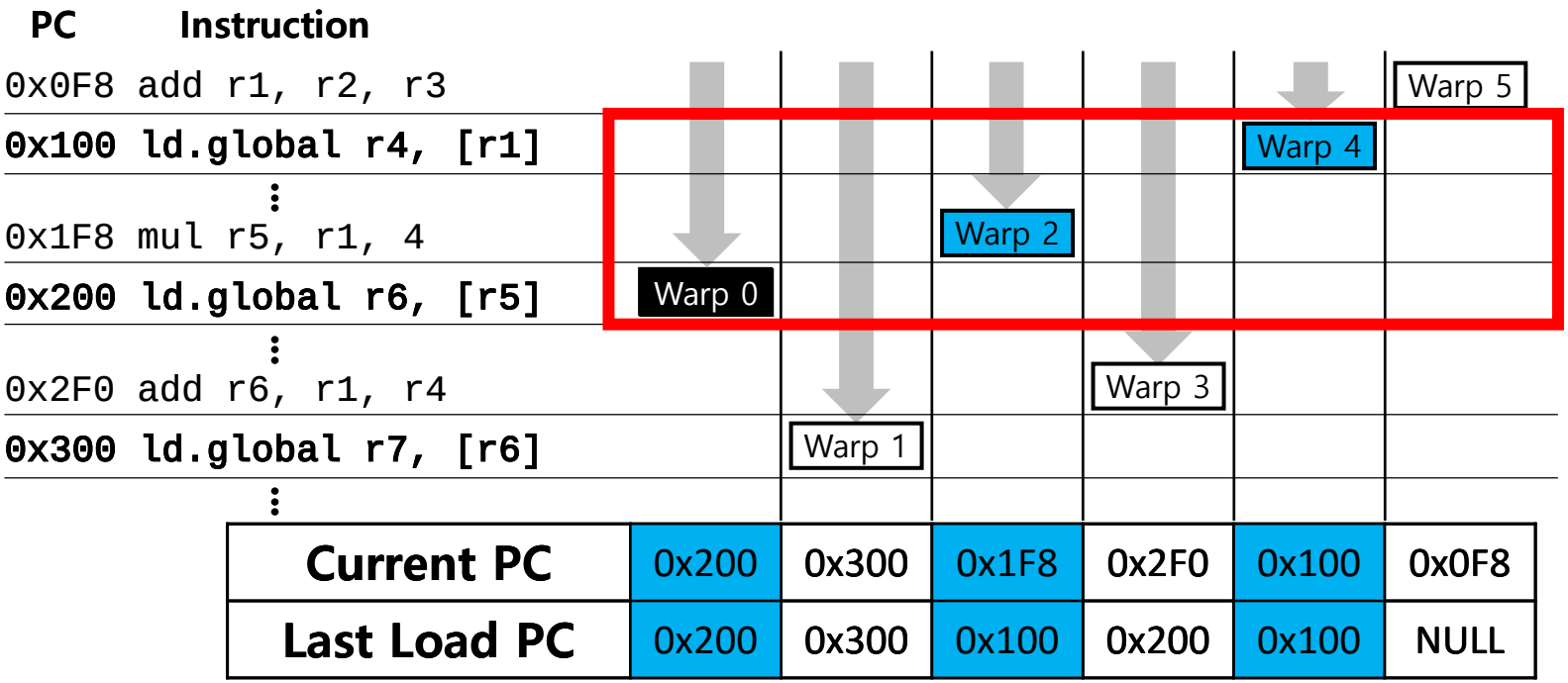
- **Locality Aware Warp Scheduler (LAWS) Groups and Prioritizes Warps.**

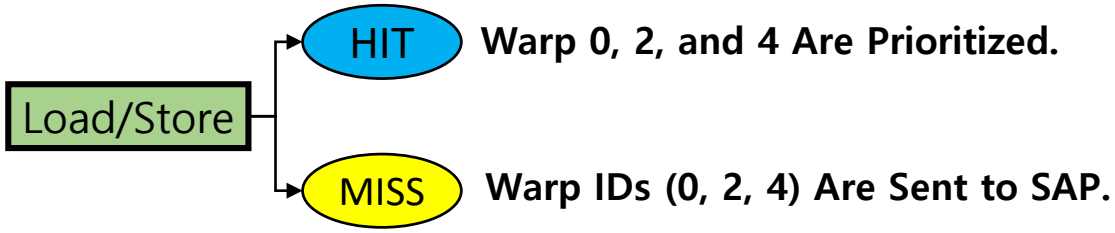
- **Scheduling Aware Prefetcher (SAP) Brings Data for Warps in a Group.**



How LAWS Works

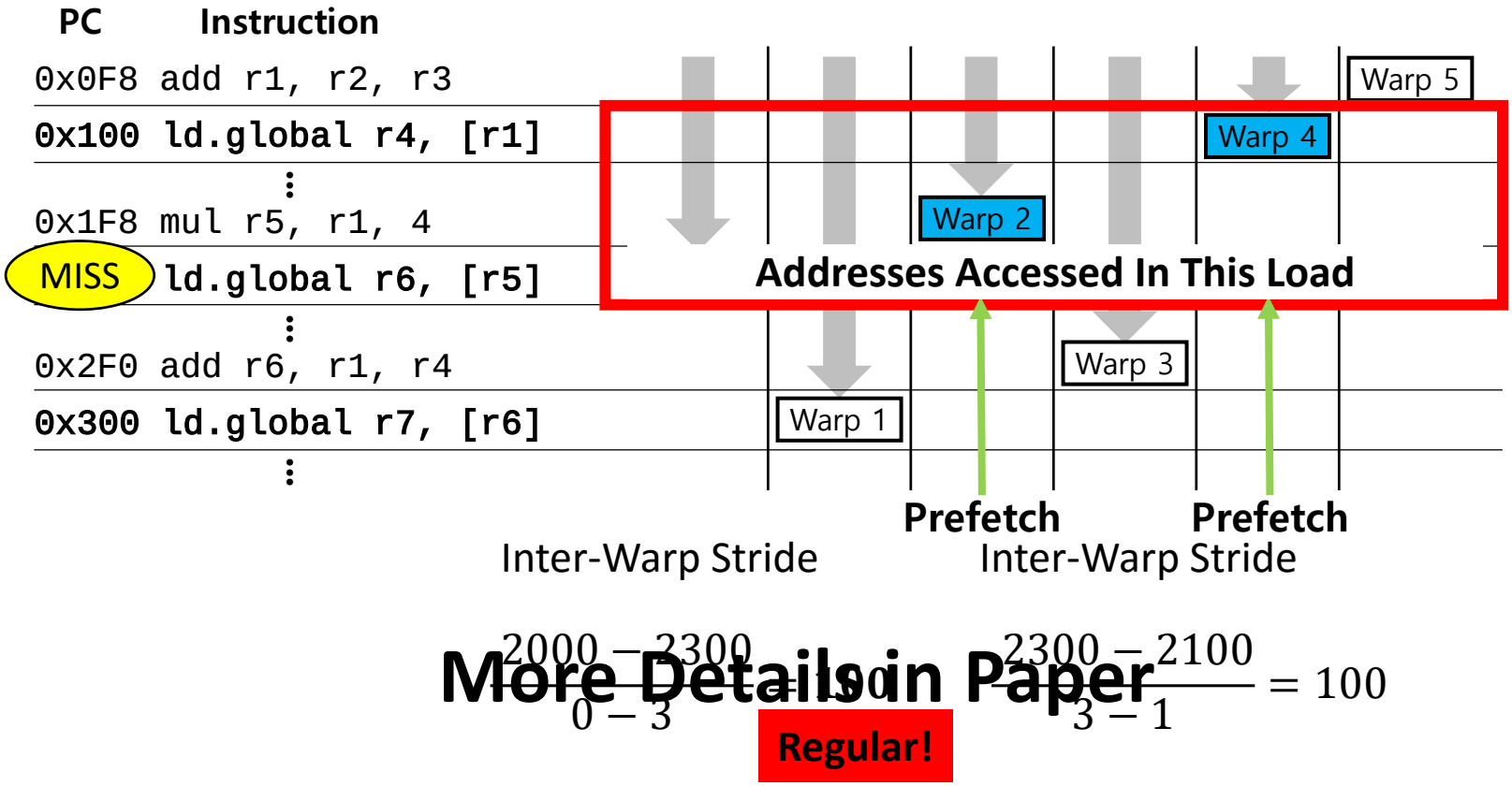
- Warps Execute Load Instructions with Round-Robin Fashion.
- LAWS Groups Warps with *"PC Address of Lastly Executed Load"*.





How SAP Works

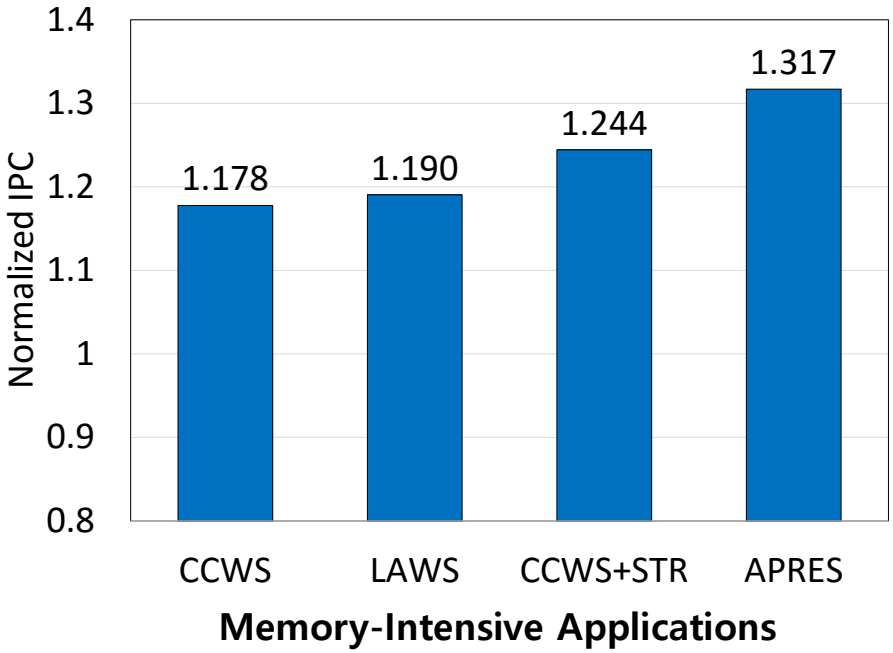
- **Inter-Warp Stride Prefetcher**
- **LAWS Prioritizes Prefetching Target Warps.**

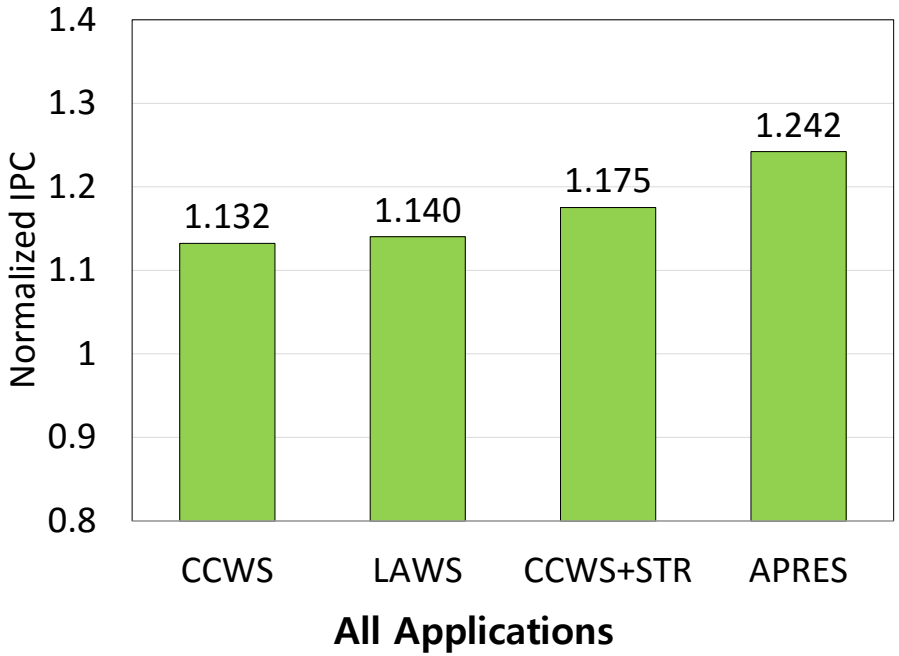


More Details in Paper

Performance of APRES

- **Performance Improvement in Memory-Intensive Applications**
  - CCWS: 17.8%, CCWS+STR: 24.4%
  - LAWS: 19.0% , APRES: 31.7%
- **24.2% of Performance Improvement in All Applications**



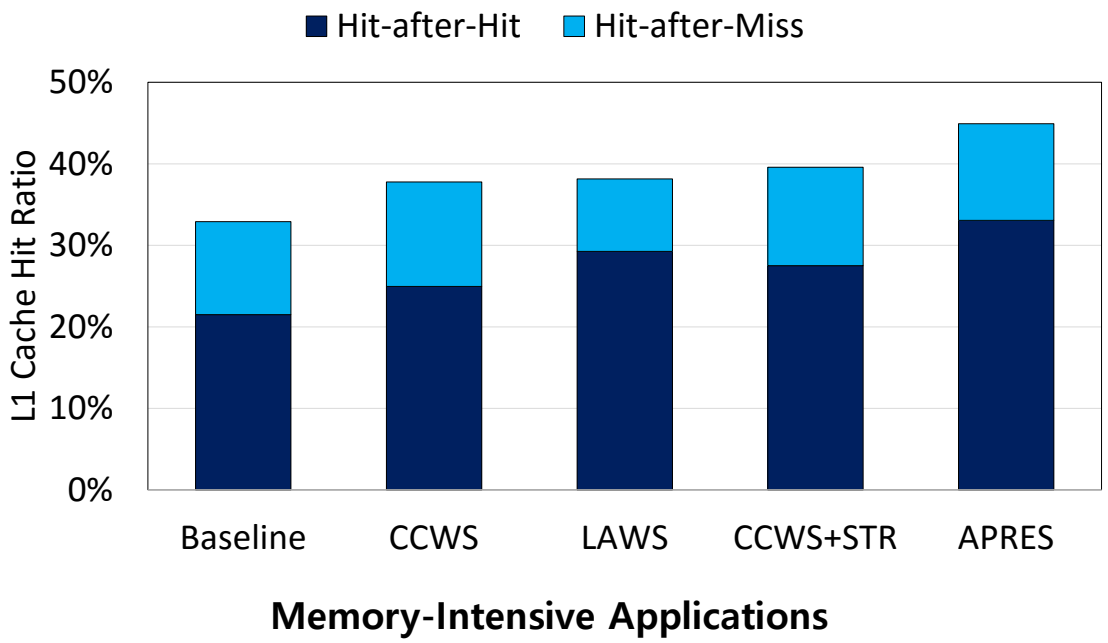


(Baseline: LRR without Prefetching)

How Much APRES Improves Cache Hits

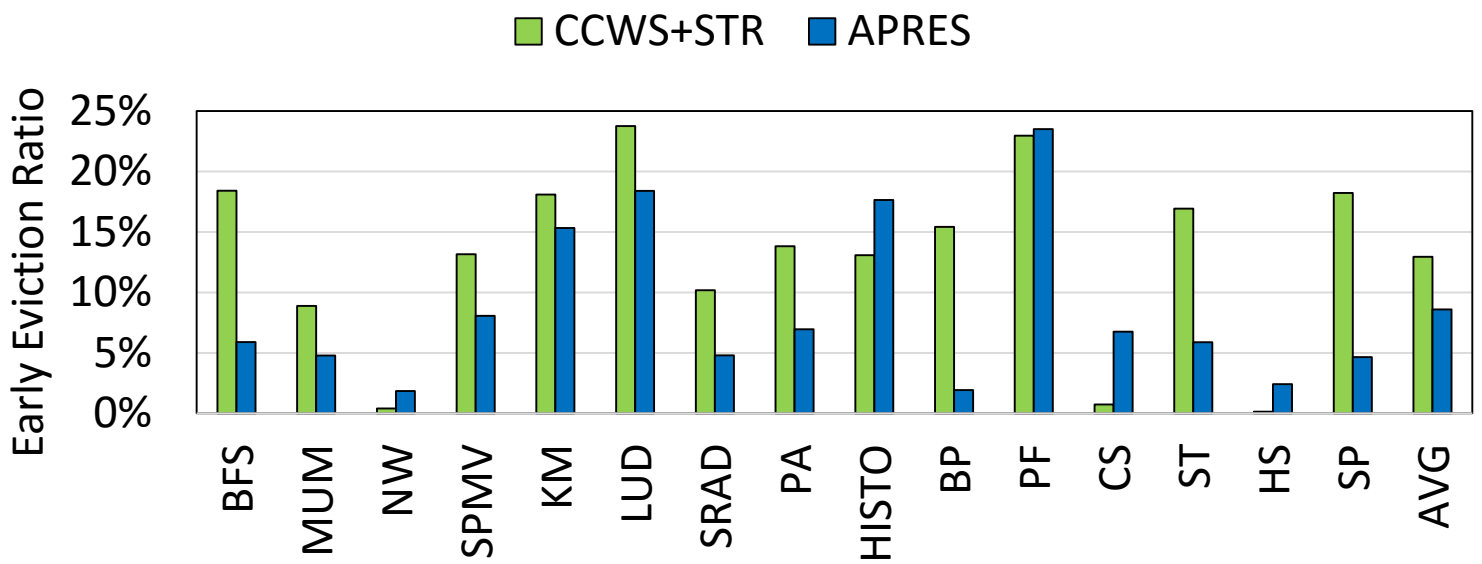
- **LAWS Makes 4.3% More Hit-after-Hit Than CCWS.**

- **APRES Makes 5.6% More Hit-after-Hit Than CCWS+STR.**



Effect on Prefetching

- **Early Eviction Ratio**
  - CCWS+STR: 13.0%
  - APRES: 8.6%



Conclusion

- **Loads with High Memory Locality are Continuously Executed by LAWS.**
- **Loads with Strided Behavior are Hit in Cache with SAP.**
- **APRES Improves Cache Efficiency on GPUs.**
  - Two load characteristics are properly exploited with adaptively cooperating warp scheduling and inter-warp stride prefetching.
  - 31.7% of performance improvement is achieved in memory-intensive workloads.

Thank You

Backup Slides

Evaluation Environment

- **Simulation Parameters**

Parameter	Value
Clock Frequency	1.4GHz
SMs / GPU	15
Warp Scheduling Policy	LRR/GTO/CCWS/MASCAR/PA
SIMT Lane Width	32
Max # of Warps / SM	48
Max # of Threads / SM	1536
Register File Size	128 KB
Max Registers / SM	32,768
# of Register Banks	32
Bit Width / Bank	128-bit
# of Entries / Bank	256
L2 Cache	8-way, 768 KB, 128B line, 200 cycles latency
DRAM	6-partitioned, 924MHz, 440 cycles latency

- **Benchmarks**
  - GPGPU-sim, Rodinia benchmark suite, Parboil benchmark suite

Characteristics of Loads in GPU Applications

%Load: Portion of each load among total load executions, **#L/#R**: # of unique cache lines per reference
Miss Rate: L1 data cache miss rate, **Stride**: Stride in bytes, **%Stride**: Portion of stride among total stride detected

Load with Small #L/#R: High Memory Locality

App	PC	%Load	#L/#R	Miss Rate	Stride	%Stride
BFS	0x110	51.6%	0.04	0.78	0	16.3%
	0xF0	26.4%	0.12	0.90	0	13.3%
	0x198	9.5%	0.11	0.83	0	14.7%
MUM	0x7A8	66.2%	0.01	0.17	0	36.3%
	0x460	21.3%	0.04	0.04	0	46.8%
	0x8A0	12.3%	0.07	0.17	0	34.3%
SPMV	0x1E0	51.5%	0.13	0.32	0	24.0%
	0x200	23.8%	0.25	0.25	0	19.3%
	0xE0	7.2%	0.65	0.81	0	12.5%
KM	0xE8	100.0%	0.03	0.99	4352	78.2%
PA	0x2210	51.7%	0.03	0.98	8832	42.7%
	0x2230	39.9%	0.002	0.16	0	36.2%
	0x2088	3.2%	0.02	0.02	256	91.5%

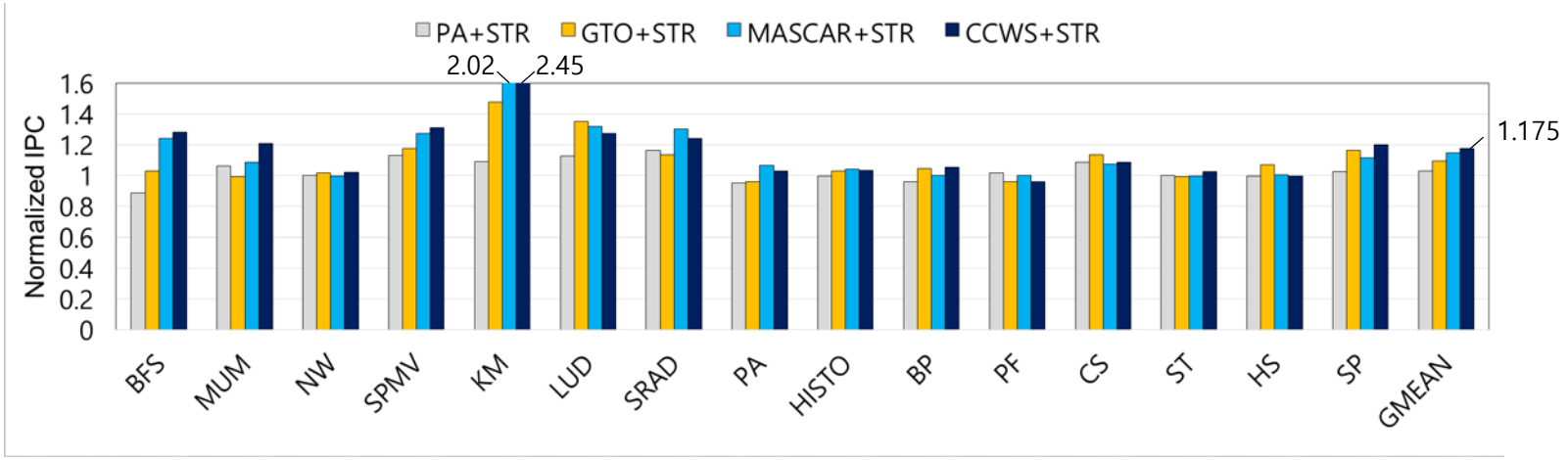
Characteristics of Loads in GPU Applications

Loads with Regular Strides

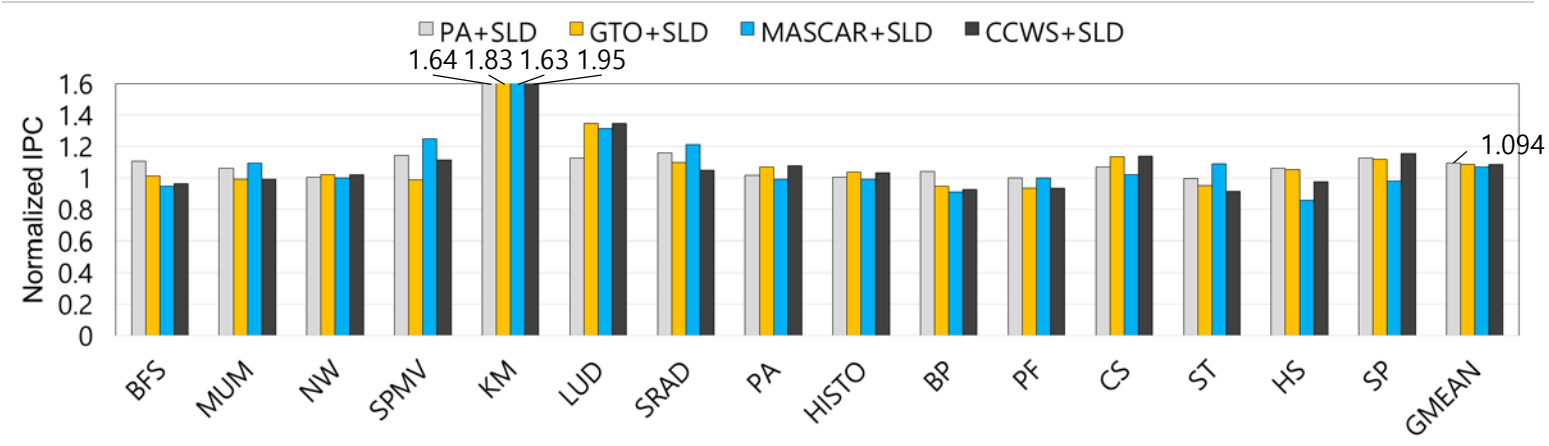
App	PC	%Load	#L/#R	Miss Rate	Stride	%Stride
NW	0x490	18.9%	0.98	1.0	-1966080	16.3%
	0xD18	18.8%	0.97	1.0	-1966080	13.3%
	0x108	1.8%	0.94	1.0	-1966080	14.7%
LUD	0x20F0	30.2%	0.58	0.32	2048	66.6%
	0x2080	30.2%	0.57	0.25	2048	83.3%
	0x22E0	30.1%	0.66	0.81	2048	77.3%
SRAD	0x250	31.2%	0.99	0.99	16384	78.2%
	0x230	31.2%	0.99	1.0	16384	75.0%
	0x350	31.2%	0.52	0.99	16384	80.7%
KM	0xE8	100.0%	0.03	0.99	4352	78.2%
PA	0x2210	51.7%	0.03	0.98	8832	42.7%
	0x2230	39.9%	0.002	0.16	0	36.2%
	0x2088	3.2%	0.02	0.02	256	91.5%
BP	0x3F8	19.4%	0.59	1.0	128	75.5%
	0x408	19.4%	0.59	1.0	128	64.1%
	0x478	19.4%	0.59	0.03	128	67.1%

Performance of STR and SLD

- **STR prefetching**

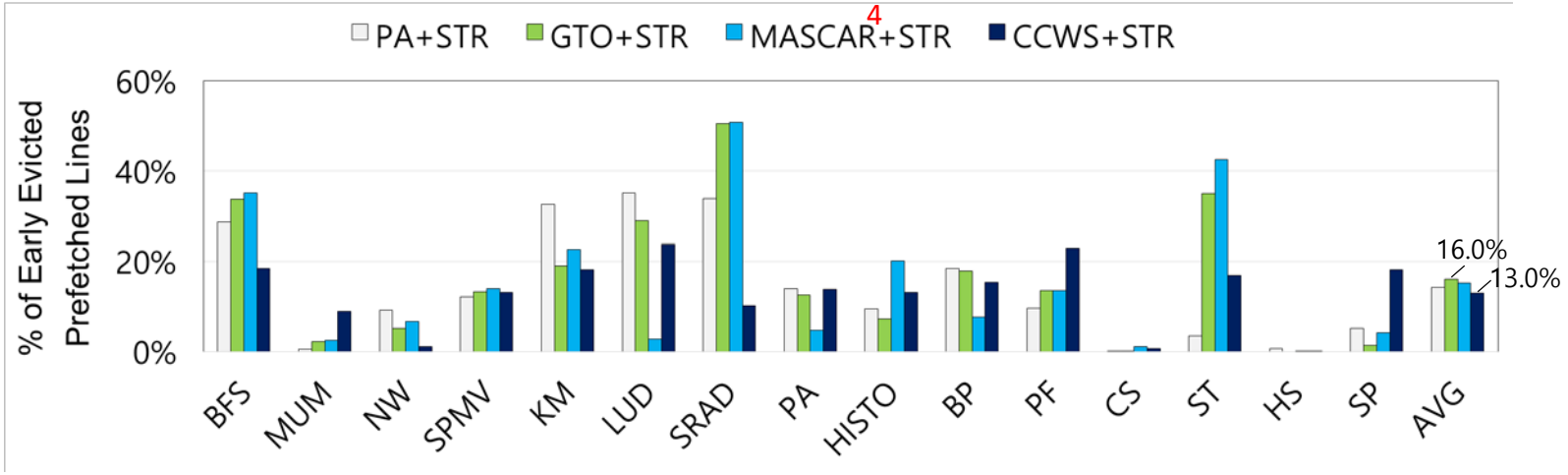


- **SLD prefetching**



Motivations: Prefetching on GPUs

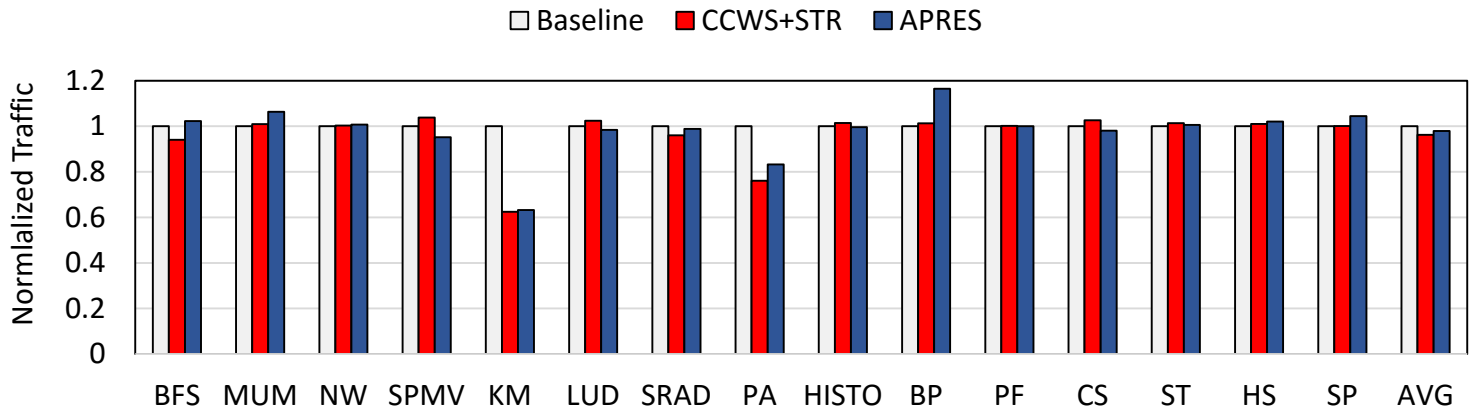
- **Previously Proposed Techniques on GPUs**
  - Stride-based prefetching (STR)[1]
  - Spatial Locality Detection (SLD) based prefetching[2]
- **Performance Improvement (with Advanced Warp Scheduler)**
  - STR: 17.5% (with Cache-Conscious Wavefront Scheduler[3])
  - SLD: 9.4% (with Prefetch-Aware Scheduler[2])
- **Early Eviction Problem**
  - Delays other prefetching and demand requests



1. Lee et al. Many-Thread Aware Prefetching Mechanisms for GPGPU Applications. MICRO 2010
2. Jog et al. Orchestrated Scheduling and Prefetching for GPGPUs. ISCA 2013
3. Rogers et al. Cache-Conscious Wavefront Scheduling. MICRO 2012
4. Sethia et al. MASCAR: Speeding up GPU Warps by Reducing Memory Pitstops. HPCA 2015

Data Traffic Comparison

- **Prefetching Techniques on GPUs Do Not Utilize Excessive Bandwidth.**
  - APRES reduces average traffics by 2.1%.
  - CCWS+STR reduces traffics by 3.8%.



Dynamic Energy Consumption

- **Total Energy Consumption Is Reduced.**
  - Power overhead of APRES is less than 3%.
  - APRES reduces energy consumption by 10.8%.