Run-Time Technique for Simultaneous Aging and Power Optimization in GPGPUs

Xiaoming Chen[1], Yu Wang[1], Yun Liang[2], Yuan Xie[3], Huazhong Yang[1]

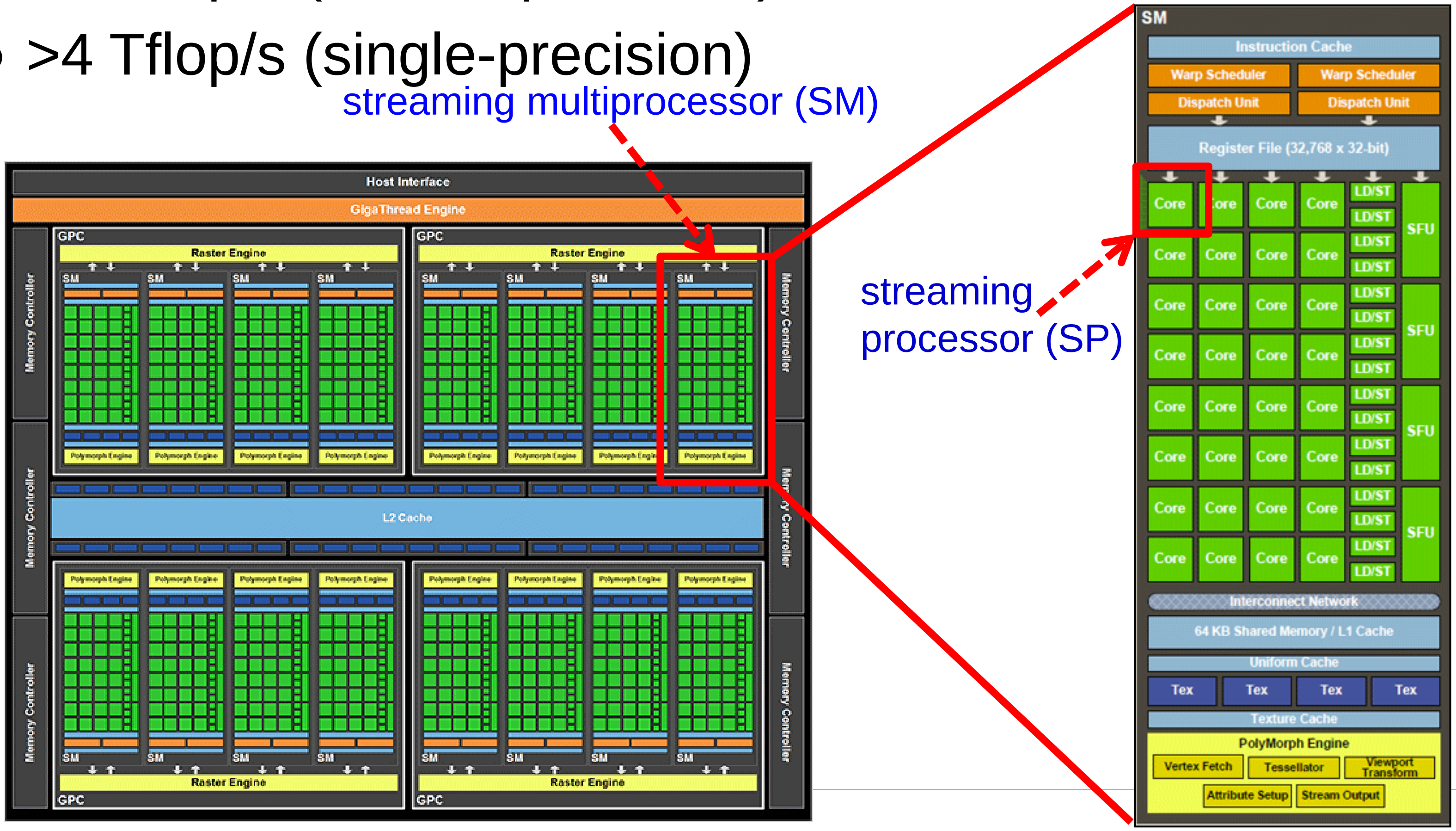
[1]Department of EE, Tsinghua University, Beijing, China

[2]School of EECS, Peking University, Beijing, China

[3]Department of CSE, Pennsylvania State University, Pennsylvania, USA

Introduction

- Massively parallel architecture: > 1000 SPs
- High performance
  - >1 Tflop/s (double-precision)
  - >4 Tflop/s (single-precision)



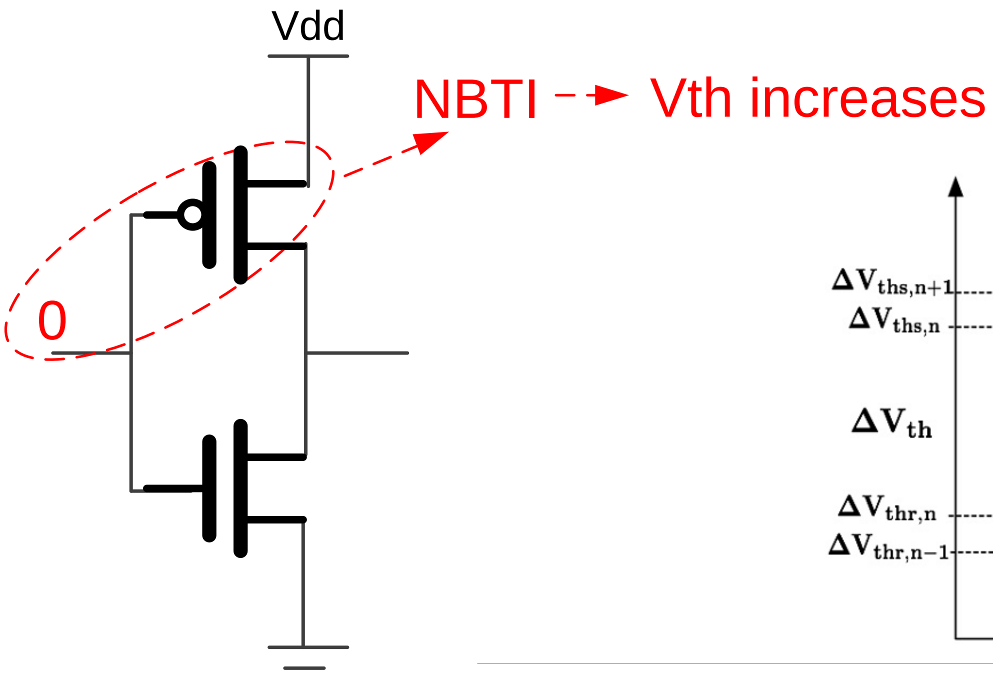
Introduction

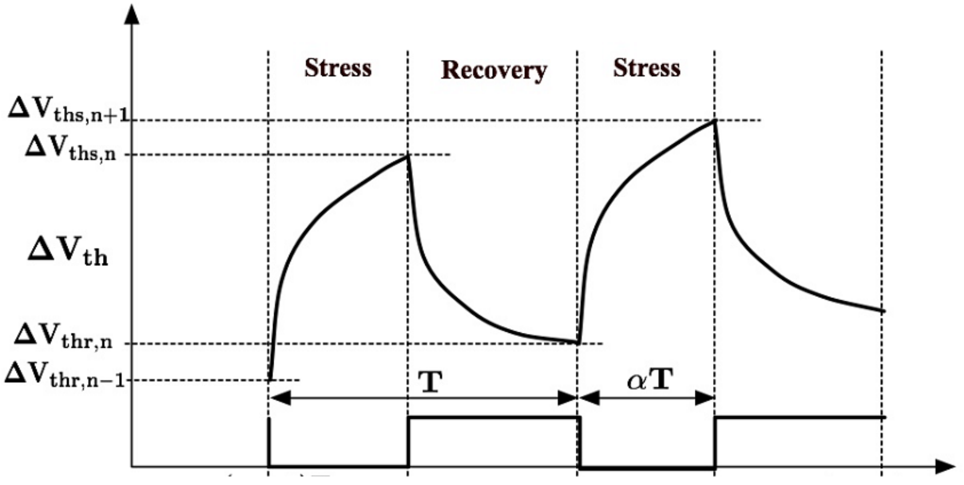
- Power is the first-order constraint for GPUs
  - Power of modern high-performance GPUs: ~250W
    - High energy consumption
    - High requirements of chip cooling techniques
    - Reliability problems

GPU	Year	Power
NVIDIA GTX580	2010	244 W
NVIDIA GTX690	2012	300 W
NVIDIA TITAN	2013	250 W
NVIDIA K40	2013	235 W
AMD 7970	2012	225 W
AMD 7990	2013	300 W

Introduction

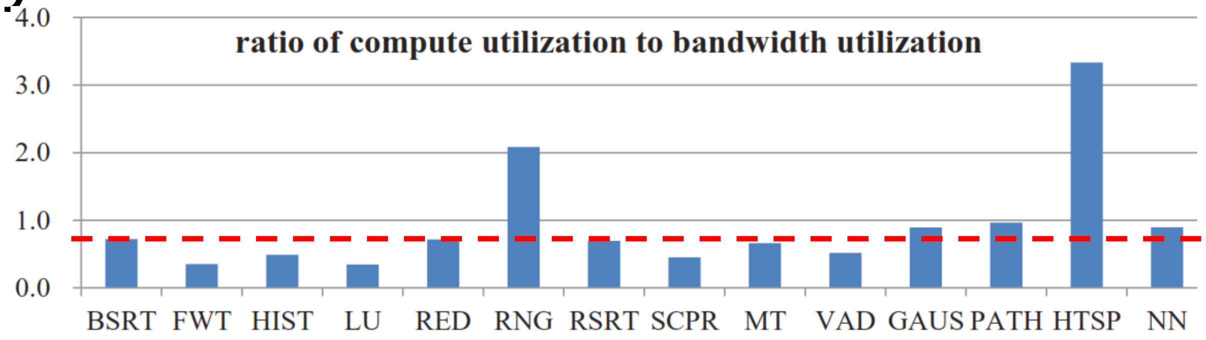
- Another challenge of modern ICs: aging effect
  - Negative bias temperature instability (NBTI): the major aging issue in nano-scale ICs
  - 20%-30% performance degradation after 3 years @ 45nm, 32nm, 22nm [Roy, D&T 2013]





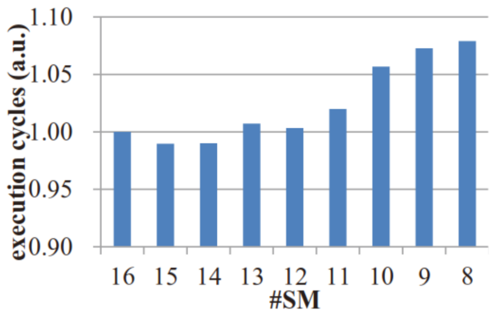
Motivation

- Low utilization of compute resources when running memory-intensive/bandwidth-bound kernels

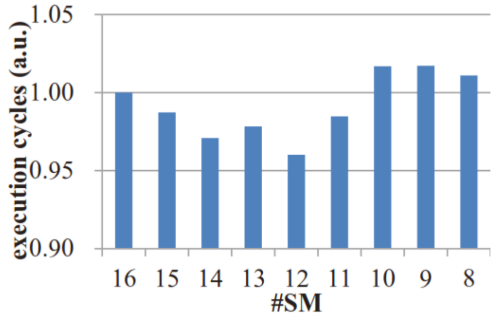


  - Compute resources often idle (waiting for memory)

- The more SMs the better?



(a) benchmark LU

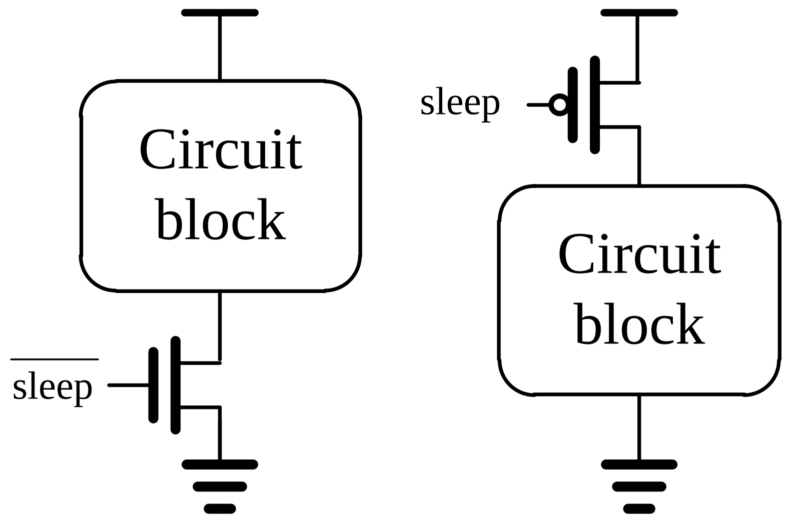


(b) benchmark RNG

  - Off-chip memory bandwidth is saturated, increasing SMs cannot improve performance

Motivation

- Power-gate some SMs when running memory-intensive/bandwidth-bound kernels
  - Power saving
  - NBTI recovery
  - Low overhead



- Key problem: what is the optimal number of SMs for a given kernel?
  - It depends on the inputs, cannot be obtained offline

Contributions

- A run-time framework for simultaneous aging and power optimization for GPGPUs

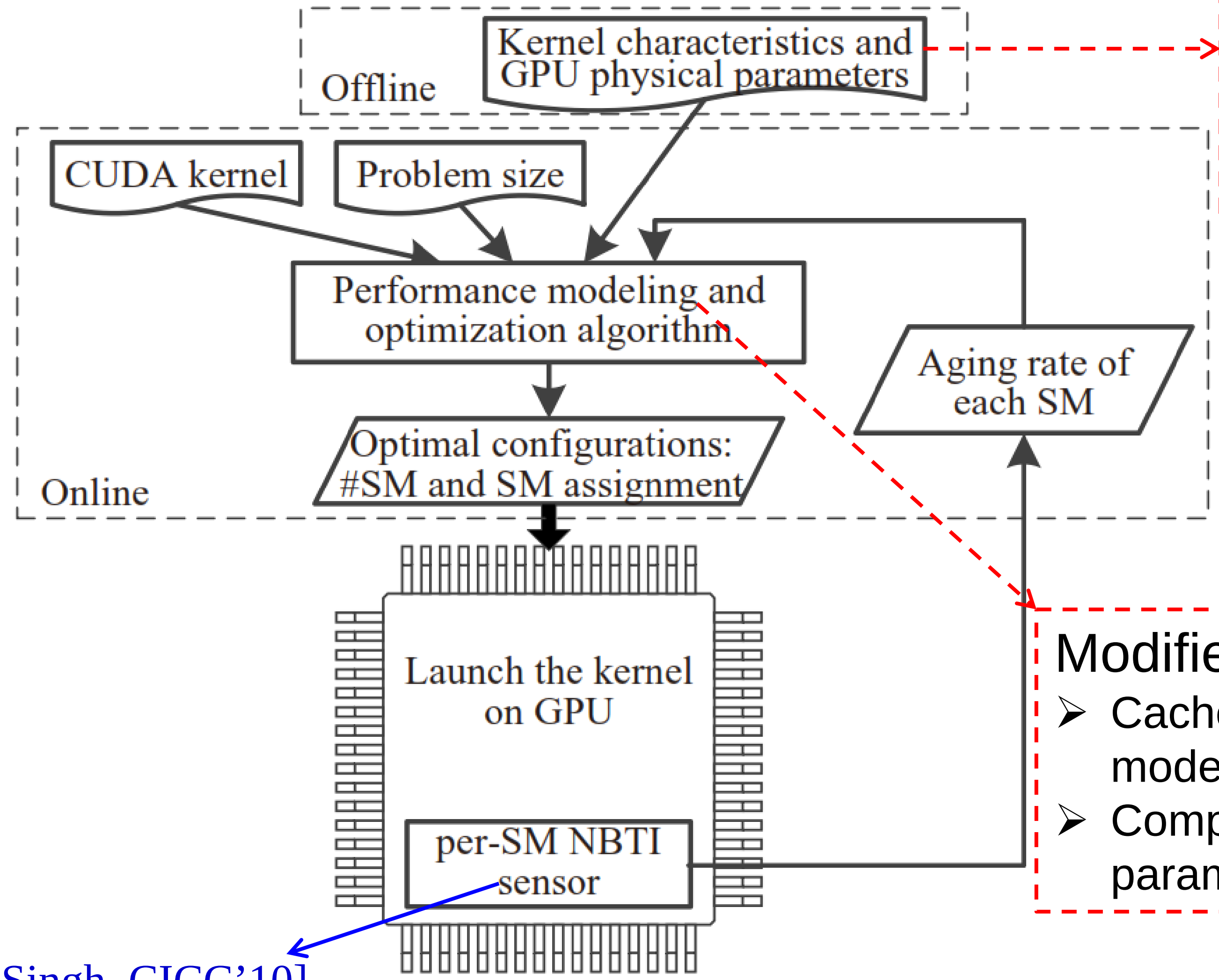
  - Observation: memory-intensive/bandwidth-bound kernels achieve the best performance with only a portion of SMs
    - The off-chip memory bandwidth is saturated

  - Method: shut down some SMs at run-time
    - A modified performance model is used to predict the optimal number of SMs online before executing a kernel

  - Effect: power reduction and aging mitigation

Our solution

- Run-time framework


Offline
Kernel characteristics and GPU physical parameters
Online
CUDA kernel
Problem size
Performance modeling and optimization algorithm
Aging rate of each SM
Optimal configurations: #SM and SM assignment
Launch the kernel on GPU
per-SM NBTI sensor

Compute cycles, memory cycles… Frequency, bandwidth, global memory latency, cache latency…

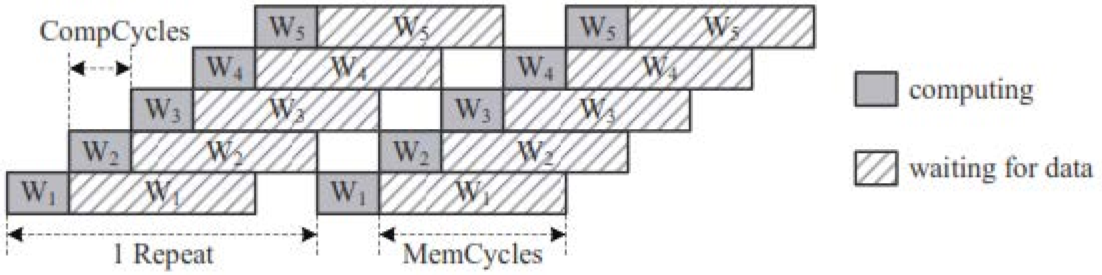
[Singh, CICC'10]

Modified from [Hong, ISCA'09]

- Cache effect is added into the model
- Computation of some parameters is also modified

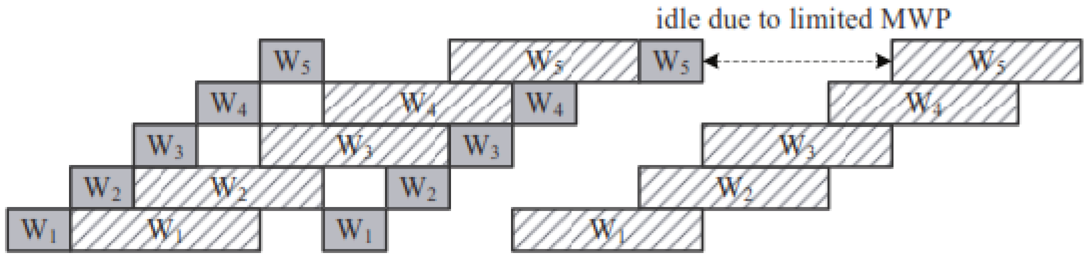
Our solution

- Performance model
  - The bandwidth is not saturated



$$TotalCycles \approx MemCycles + Repeat \times WarpsPerSM \times CompCycles$$

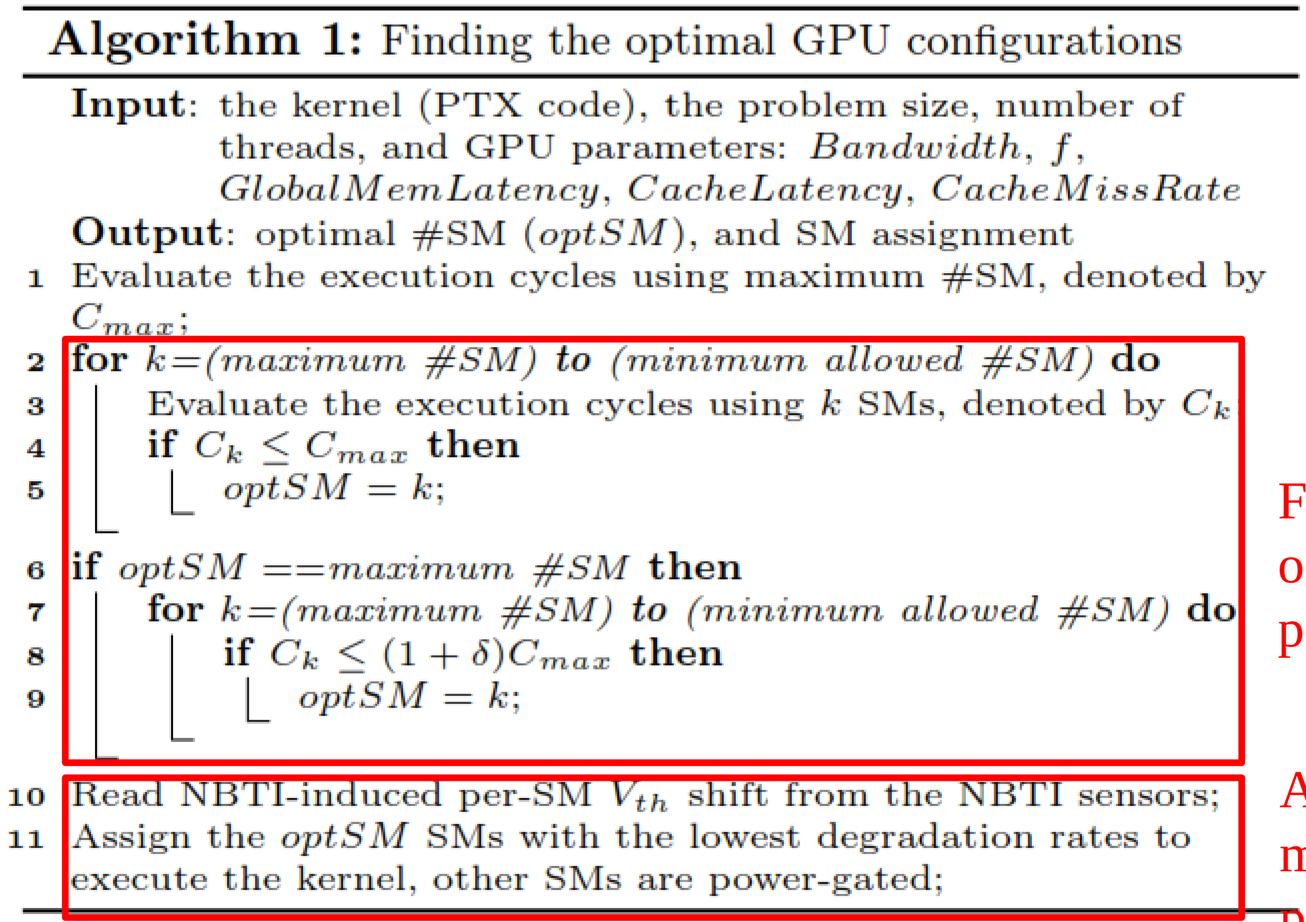
  - The bandwidth is saturated



$$TotalCycles \approx Repeat \times \frac{WarpsPerSM}{MWP} \times MemCycles$$

Our solution

- Online optimization algorithm



Algorithm 1: Finding the optimal GPU configurations

```
Input: the kernel (PTX code), the problem size, number of
       threads, and GPU parameters: $Bandwidth$, $f$,
       $GlobalMemLatency$, $CacheLatency$, $CacheMissRate$
Output: optimal #SM ($optSM$), and SM assignment
1  Evaluate the execution cycles using maximum #SM, denoted by
   $C_{max}$;
2  for $k$=($maximum$ $\#SM$) to ($minimum$ $allowed$ $\#SM$) do
3      Evaluate the execution cycles using $k$ SMs, denoted by $C_k$
4      if $C_k \leq C_{max}$ then
5          $optSM = k$;
6  if $optSM$ ==$maximum$ $\#SM$ then
7      for $k$=($maximum$ $\#SM$) to ($minimum$ $allowed$ $\#SM$) do
8          if $C_k \leq (1+\delta)C_{max}$ then
9              $optSM = k$;
10 Read NBTI-induced per-SM $V_{th}$ shift from the NBTI sensors;
11 Assign the $optSM$ SMs with the lowest degradation rates to
   execute the kernel, other SMs are power-gated;
```

Find the optimal number of SMs through the performance model

Assign SMs with the minimum aging rate, power gate other SMs

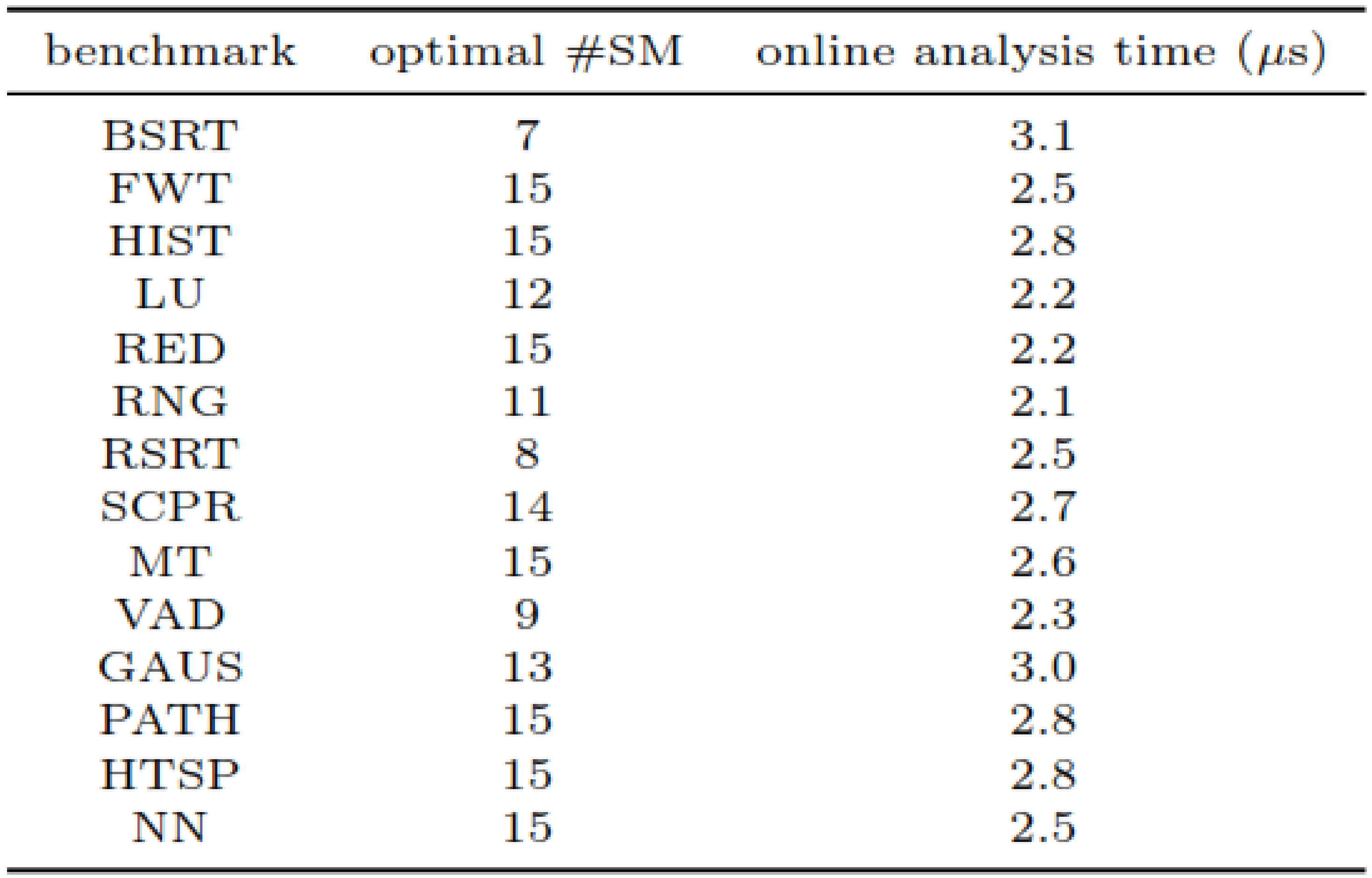
Experimental setup

- **Benchmarks**
  - CUDA SDK example
  - Rodinia [Che, IISWC’09]
  - Real-world kernels
- **Performance and power** evaluation: GPGPU-Sim [Bakhoda, ISPASS’09] with GPUWattch [Leng, ISCA’13]
- **NBTI** evaluation: NBTI analytical model [Bhardwaj, CICC’06]
- **Temperature** evaluation: Hotspot [Huang, ISPASS’09]
  - For NBTI calculation
- Baseline: GPU (16 SMs) without power gating

Simulation results

- Optimal #SM and analysis time

Table 2: Results of the optimization algorithm.



benchmark	optimal #SM	online analysis time (μs)
BSRT	7	3.1
FWT	15	2.5
HIST	15	2.8
LU	12	2.2
RED	15	2.2
RNG	11	2.1
RSRT	8	2.5
SCPR	14	2.7
MT	15	2.6
VAD	9	2.3
GAUS	13	3.0
PATH	15	2.8
HTSP	15	2.8
NN	15	2.5

Simulation results

- Performance degradation: < 1%
  - Caused by shutting down some SMs and the online optimization algorithm
- Power reduction: 19%
- Energy reduction: 18%
- Reduction in NBTI-induced Vth shift: 34%

Simulation results

- Our technique is implemented at run-time, it can handle different problem sizes

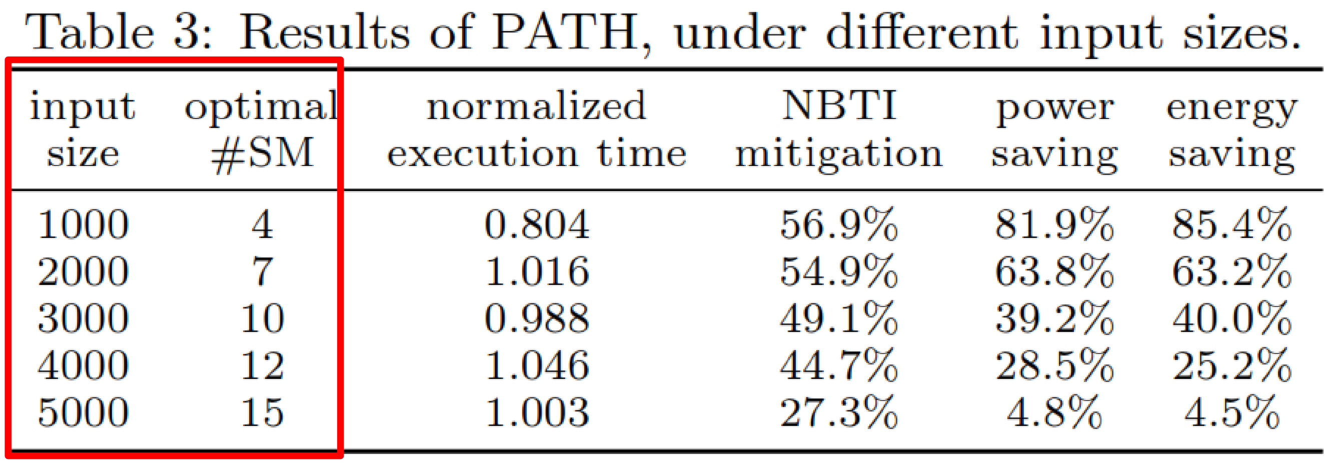


Table 3: Results of PATH, under different input sizes.

input size	optimal #SM	normalized execution time	NBTI mitigation	power saving	energy saving
1000	4	0.804	56.9%	81.9%	85.4%
2000	7	1.016	54.9%	63.8%	63.2%
3000	10	0.988	49.1%	39.2%	40.0%
4000	12	1.046	44.7%	28.5%	25.2%
5000	15	1.003	27.3%	4.8%	4.5%

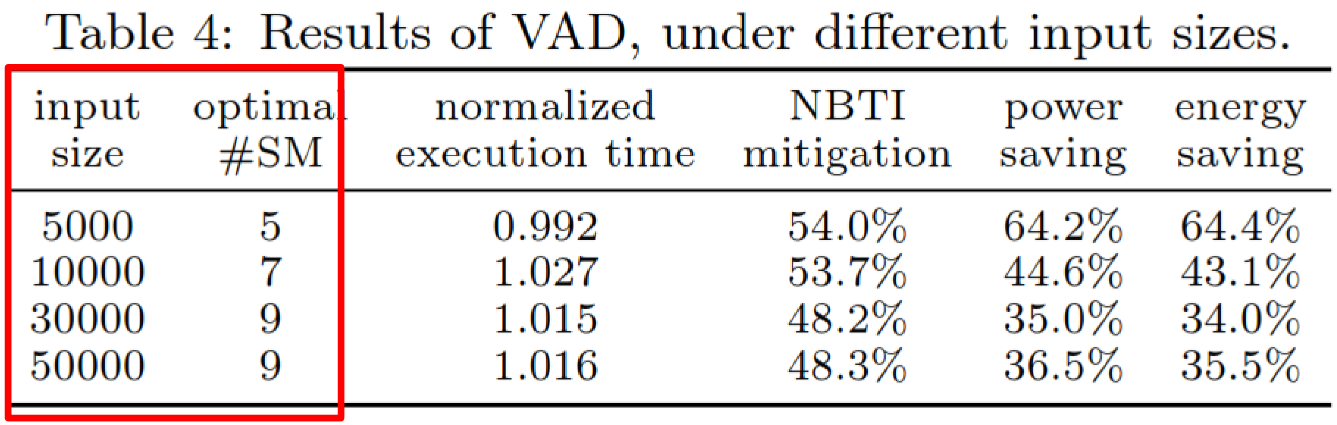


Table 4: Results of VAD, under different input sizes.

input size	optimal #SM	normalized execution time	NBTI mitigation	power saving	energy saving
5000	5	0.992	54.0%	64.2%	64.4%
10000	7	1.027	53.7%	44.6%	43.1%
30000	9	1.015	48.2%	35.0%	34.0%
50000	9	1.016	48.3%	36.5%	35.5%

Conclusions

- Memory-intensive/bandwidth-bound kernels do not need all the compute resources
  - Memory bandwidth is saturated when using a portion of SMs
- A predictive shutting down framework to perform power gating for SMs in GPUs
  - A modified performance model is used to predict the optimal number of SMs
  - Assign SMs with the minimum aging rate, power gate other SMs
  - NBTI mitigation and power reduction are both achieved

Thanks for your attention

Q & A