GPGPU Workload Characteristics and Performance Analysis

Sohan Lal, Jan Lucas, Michael Andersch
Mauricio Alvarez-Mesa, Ahmed Elhossini, Ben Juurlink
Embedded Systems Architecture
TU Berlin, Einsteinufer 17, D-10587 Berlin, Germany
http://www.aes.tu-berlin.de
{sohan.lal, j.lucas, michael.andersch, alvarez, ahmed.elhossini, juurlink}@aes.tu-berlin.de

***Abstract*—GPUs are much more power-efficient devices compared to CPUs, but due to several performance bottlenecks, the performance per watt of GPUs is often much lower than what could be achieved theoretically. To sustain and continue high performance computing growth, new architectural and application techniques are required to create power-efficient computing systems. To find such techniques, however, it is necessary to study the power consumption at a detailed level and understand the bottlenecks which cause low performance. Therefore, in this paper, we study GPU power consumption at component level and investigate the bottlenecks that cause low performance and low energy efficiency. We divide the low performance kernels into low occupancy and full occupancy categories. For the low occupancy category, we study if increasing the occupancy helps in increasing performance and energy efficiency. For the full occupancy category, we investigate if these kernels are limited by memory bandwidth, coalescing efficiency, or SIMD utilization.**

I. Introduction

It has not been even a decade since GPUs entered the mainstream computing domain, but they have already made quick inroads into many domains including the high performance computing. The main reason is the tremendous computing power offered by GPUs which is increasing with every new generation. GPUs are massively multi-threaded, throughput oriented devices that employ huge number of parallel threads to achieve high throughput. The peak throughput of GPUs is a magnitude higher than CPUs. The higher throughput also comes with higher power consumption. However, GPUs are more power-efficient devices [1] compared to CPUs, as performance per watt of GPUs is much higher than CPUs. For example, NVIDIA's GTX 690 has 18.7 SP GFLOPS/W while Intel's Haswell i7 4770K has 5.3 SP GFLOPS/W. However, due to various performance bottlenecks which results in under-utilization of resources, the performance per watt of GPUs is often much lower than what could be gained theoretically. There are several factors that contribute to low performance, including low occupancy, memory bandwidth, control flow divergence, and memory divergence.

To create power-efficient techniques at the architectural level, we need to gain GPUs power consumption knowledge at a fine-grained level and understand the bottlenecks to low performance [2], [3]. Therefore, in this paper we study GPU power consumption at the component level for a diverse set of workloads and investigate the bottlenecks which cause low performance and power efficiency. We explore correlation between workload metrics such as IPC and SIMD utilization and components power consumption to understand how workload characteristics affect power consumption. Moreover, the workloads are studied at the kernel level rather than benchmark level.

To investigate the bottlenecks for low performance, we divide the low performance kernels into two categories: low occupancy and full occupancy. The low occupancy kernels are further divided into different categories depending upon the resources their occupancy is limited by. We increase the occupancy of each category by increasing the corresponding resources and study if high occupancy helps in achieving higher performance and energy efficiency. We show that increasing the occupancy helps in increasing performance and energy efficiency for most of the kernels, but just increasing occupancy is not enough to achieve the maximum performance. The full occupancy kernels are analyzed for memory bandwidth utilization, coalescing efficiency, and SIMD utilization. We show which kernels are limited by memory bandwidth or low coalescing efficiency or low SIMD utilization or any combination of these.

We make the following contributions:

- We study GPUs power consumption at the component level and investigate their correlation with workload metrics.
- We investigate the bottlenecks of low performance category and study if increasing the occupancy helps in increasing the performance.
- We also analyze the kernels having full occupancy but still performing low and study if these kernels are limited by memory bandwidth, low coalescing efficiency or low SIMD utilization.

The rest of the paper is organized as follows. Section II describes related work. Section III presents GPUs power efficiency and our bottlenecks investigation methodology. In the Section IV we explain experimental setup. Section V presents investigation results. Finally, we draw conclusions in Section VI.

978-1-4799-3770-7/14/$31.00 ©2014 IEEE

II. Related Work

Related work for this paper can be divided into two categories. First, the previous work done for components power consumption of GPUs and their correlation with workload metrics and second, the work done for the bottlenecks analysis of low performance workloads. There are some works which estimate GPUs power consumption, but they do power estimate at a very coarse-grained level. Ma et al. [4] used statistical analysis to develop GPU power consumption model and reported power consumption for entire GPU for few benchmarks. Gebhart et al. [5] used a very simple and high level power model to estimate the total core power. According to their work, cores consume up to 70% of total power, but this is not enough for power optimization as we need to understand power consumption at much fine-grained level. In contrast to this, we study power consumption at component level. The recent release of GPUSimPow [6] and GPUWattch [7], GPUs power estimation tools has enabled in-depth exploration of GPUs power consumption. Using metrics to understand workload characteristics is not new [8], [9], [10], [11], but none of them study the correlation between workload metrics and components power. We compute the correlation between workload metrics and components power to understand the power characteristics of various workloads. In addition to this, we also quantify the change in components power consumption with the change in workload characteristics.

Blem et al. [12] characterized a set of benchmarks to find their performance bottlenecks and predict the performance improvements after mitigating those bottlenecks. We also investigate the bottlenecks of low performance workload, but there are key differences both in the methodology and performance metrics used. First, we use a performance simulator not only for bottlenecks identification but also for performance prediction, unlike Blem et al. [12]. They use analytical model to predict performance, which according to their work has error in the range -70% to 2×, which is high and a limitation of their work. Second, we also report power and energy changes. Third, they observe that low available parallelism is a bottleneck but do not consider the case that even with high available parallelism, actual parallelism (occupancy) could still be very low and hence, low performance. We show that a large number of kernels have low occupancy and how increasing the occupancy helps increasing performance and energy efficiency.

III. GPUs power efficiency and performance bottlenecks

GPUs are power-efficient devices at full utilization. All top ten supercomputers in the green 500 list contains GPUs (www.green500.org). However, due to various bottlenecks which results in under-utilization of resources, the performance per watt of GPUs is much lower than what could be gained at full utilization. Energy per instruction (E/I) and IPC per watt (IPC/W) are metrics often used to study the power efficiency of GPU workloads. Figure 1 shows the E/I (nJ) and IPC/W against IPC for several kernels. Each point represents a kernel. For a description of benchmarks used please refer to Section IV-C. The figure shows that the E/I increases with the decrease in IPC which results in low power efficiency for kernels with low IPC. The exact cut for low IPC may be a point of open discussion, but the trend shows that the lower IPC results in lower power efficiency. For this study, we classify the kernels with IPC >50.5% of peak IPC into *high performance* (HP) category and kernels with IPC <= 50.5% of peak IPC into *low performance* (LP) category. We choose the cut at 50.5% instead of 50% because there is one kernel with IPC 50.5% and it lies closer to LP category than HP category. The peak IPC is 1024 (#SM×#FU per SM×2 = 16×32×2) for the simulator configuration described in Table II. There is a factor of 2 in the formula because gpgpu-sim simulates the full warp at half frequency [13].

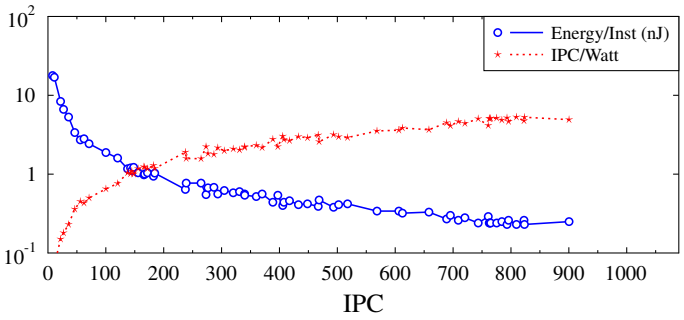


Fig. 1: Power efficiency.

The average E/I for the HP and LP category is 0.27 (nj) and 2.01 (nj), respectively. The later is 7.5× less energy efficient compared to former, a huge difference which is not good for the future growth of high performance computing. The HP and LP category has 21 and 47 kernels, respectively and the average IPC for the former is 741 and 250 for the later, which is less than 25% of the peak IPC. Surprisingly, more than 69% of the kernels belongs to LP category. Blem et al. [12] also noticed that over half of the benchmarks they studied have IPC less than 40% of the peak IPC for Tesla C1060. The figure also shows that the IPC/W decreases with the decrease in IPC. The average IPC/W for HP and LP category is 4.61 and 1.65 respectively. Thus, the LP category has low performance and power efficiency. In the following section, we describe our methodology to investigate the reasons for low performance.

A. Bottlenecks Investigation

GPUs are high throughput devices and use large number of threads to hide the long latency operations. Occupancy is the metric used to measure the number of threads allocated to a streaming multiprocessor (SM) of a GPU. It is defined as a ratio of threads allocated to an SM and the maximum number of threads that can be allocated to an SM. A certain minimum occupancy, which may vary from kernel to kernel depending on ILP, ratio of arithmetic to memory operations etc., is necessary to hide latency and to achieve high throughput. The occupancy depends on parallelism in a kernel, the resources requested by the kernel and the resources available on the GPU.

We do not consider those kernels for low performance analysis which do not have enough parallelism and hence also not enough threads to fill all the SMs. We argue that although it is possible to get higher performance with lower occupancy [14], but we only consider the case where parallelism is not an issue but other architectural resources are bottleneck to higher performance and energy efficiency. The resources requested by a kernel are allocated for the entire CTA (Cooperative Thread Arrays in NVIDIA terminology) and at least one CTA needs to be allocated for the GPU to work. A CTA is a group of concurrent threads that execute the same thread program and may cooperate via shared memory to compute results. GPU may not have the requested resources to allocate enough CTAs to fully occupy the GPU which results in low occupancy. Table I shows the resources constraint for full occupancy on NVIDIA's GTX580 which can hold maximum of 1536 threads per SM. Any kernel which has less than 192 threads/CTA or require more than 21 registers/thread or more than 6KB shared memory/CTA cannot have full occupancy. Thus, occupancy can be limited by CTAs limit, registers usage, and shared memory usage.

Resource	Max	Required for full occupancy
CTA limit	8	Min 192 threads/CTA
Registers	32K	Max 21/thread
Shared memory	48KB	Max 6KB/CTA

TABLE I: Resources constraint for full occupancy.

We divide the LP category kernels into two categories for further analysis: low occupancy and full occupancy. The low occupancy category kernels have occupancy <1 and full occupancy category kernels have occupancy = 1. The low occupancy could restrict the latency hiding capabilities of the GPU and hence could restrict performance as well. We investigate the effect of increasing the occupancy of such kernels and show the gain in performance and energy in the Section V-C. We further classify the low occupancy kernels depending on the resources they are limited by. The low occupancy kernels that we study fall in one of the following categories.

- Limited by CTA limit
- Limited by registers
- Limited by shared memory

The full occupancy category kernels have maximum number of threads that can be assigned to the SM, but still performing low. In such a case the most likely bottlenecks are high bandwidth utilization, low coalescing efficiency, and low SIMD utilization. For the full occupancy category, we investigate if any of these is a bottleneck for high performance in Section V-D

IV. Experimental Methodology

A. Simulator

We use the GPUSimPow simulator for simulating different benchmarks [6]. The simulator has an average relative error of 11.7% and 10.8% between simulated and hardware power for GT240 and GTX580, respectively. For more information regarding the simulator, please refer to [6]. We use GPUSimPow to simulate a GPU similar to NVIDIA's GF110 on the GTX580 card. The baseline simulator configuration is summarized in Table II.

#SMs	16	Shared memory/SM	48KB
SM freq (MHz)	822	L1 $ size/SM	16KB
Max #Threads per SM	1536	L2 $ size	768KB
Max #CTA per SM	8	# Memory controllers	6
Max CTA size	512	Memory type	GDDR5
#FUs per SM	32	Memory clock	2004 MHz
#Registers/SM	32K	Memory bandwidth	192.4 GB/s

TABLE II: Baseline simulator configuration

B. Evaluated GPU Components

We briefly describe the GPU components evaluated for power consumption. For more details of the components, its subcomponents and power model please refer to [6].

1) Register file (RF): contains multiple SRAM banks, crossbar, and operand collectors.
2) Execution Units (EU): contains integer, floating point, and special function units.
3) Warp control unit (WCU): front end of the GPU, contains warp status table, instruction buffer, reconvergence stack, and scoreboard as subcomponents.
4) Base power (BP): consumed when a SM is activated.
5) Load store unit (LSU) : handles load and store requests to memory subsystem. In our evaluated component it contains coalescer, bank conflict checker, shared memory, L2 cache, constant cache, and texture cache.
6) Clusters power (CP): consumed when a cluster is activated.
7) Network on chip (NOC): connects SMs to global memory.
8) Memory controller (MC): current generation of GPUs such as Fermi use 64-bit memory controllers.
9) Global memory (GM): current generation of GPUs such as Fermi use either GDDR3 or GDDR5 SGRAM.
10) Total Power (TP): consumed by all GPU components.

C. Benchmarks

Table III shows the benchmarks used for evaluation. The benchmark selection includes benchmarks from the popular Rodinia benchmark suite [15] and CUDA SDK [16]. In addition to Rodinia and CUDA SDK, our benchmarks selection also includes benchmarks recommended by Goswami et al. [10] and an internally developed motion compensation kernel from H264.

D. Workload metrics

We use several workload metrics to study the performance characteristics as in [10], [8]. Following is a short description of each metric.

1) IPC: Instructions per cycle.

Name	Abbreviation	#Kernels	Description	Origin
backprop	BP	2	Multi-layer perceptron training	Rodinia
bfs	BFS	2	Breadth-first search	Rodinia
b+tree	BT	2	Graph search	Rodinia
cfd	CFD	4	Computational fluid dynamics	Rodinia
heartwall	HW	1	Ultrasound image tracking	Rodinia
hotspot	HS	1	Processor temperature estimation	Rodinia
kmeans	KM	2	k-means clustering	Rodinia
lavaMD	MD	1	Molecular dynamics	Rodinia
leukocyte	LC	3	Microscopy video tracking	Rodinia
mummergpu	MUM	2	Pairwise local sequence alignment	Rodinia
pathfinder	PF	1	Dynamic programming path search	Rodinia
srad_v1	SRAD1	6	Speckle reducing anisotropic diffusion	Rodinia
srad_v2	SRAD2	2	Speckle reducing anisotropic diffusion	Rodinia
similarityScore	SS	17	Similarity score calculation	Rodinia
blackscholes	BS	1	Black-Scholes PDE solver	CUDA SDK
binomialOptions	BN	1	Binomial options pricing	CUDA SDK
convolutionSeparable	CS	2	Convolution	CUDA SDK
fastWalshTransform	FWT	3	Fourier transform	CUDA SDK
histogram	HG	4	Histograms for analysis	CUDA SDK
mergesort	MS	4	Parallel merge-sort	CUDA SDK
monteCarlo	MC	2	Monte carlo numerical solver	CUDA SDK
scalarprod	SP	1	Scalar product of two vectors	CUDA SDK
scan	SCAN	3	Parallel prefix sum	CUDA SDK
transpose	MT	8	Computation of matrix transpose	CUDA SDK
vectoradd	VA	1	Addition of two vectors	CUDA SDK
storegpu	STO	1	Distributed storage systems	Third party [17]
motionCompensation	MCO	2	H264 video decoding	Third party

TABLE III: GPGPU benchmarks used for experimental evaluation.

2) Arithmetic Inst. (AI): Ratio of arithmetic instructions to total instructions.
3) Branch Inst. (BI): Ratio of branch instructions to total instructions.
4) Memory Inst. (MI): Ratio of memory instructions to total instructions.
5) Bandwidth Utilization (BW): Ratio of bandwidth utilized and bandwidth available.
6) Coalescing Efficiency (CE): Ratio of global memory instructions and global memory transactions.
7) SIMD Utilization (SU): Average utilization of SM core for issued cycles. It does not include the cycles for which pipeline is stalled and cannot issue instructions.
8) Pipeline Stalled (PS): The fraction of total cycles where pipeline is stalled and could not issue instructions.
9) Active Warps (AW): Number of active warps per SM.

V. Results

In the Section V-A, we present correlation results. In the Section V-B, we discuss components power consumption. The bottlenecks investigation results for the low and full occupancy categories in the Sections V-C and V-D.

A. Correlation

We calculated the Pearson correlation coefficient between the workload metrics and components power consumption for all kernels. The Pearson correlation coefficient is a measure of linear dependence between the two variables and it varies between -1 and 1. Higher absolute value of correlation coefficient means strong linear dependence between the metric and the corresponding component. The negative value means there is an inverse dependence. Since the static power is caused by leakage currents and it does not depend on the workload, but only on architecture where the workload is executed, therefore, we only consider dynamic power for studying the correlation and components power consumption in Section V-B.

We found that IPC has strong correlation with RF (0.95), EU (0.92), and WCU (0.80). The metrics related to types of instructions AI, BI, and MI do not have strong correlation with any of the components, but shows some expected trends. For example, AI has positive correlation with RF (0.29), EU (0.42), WCU (0.15), but it has negative correlation with MC (-0.02). BW utilization has very strong correlation with MC (0.98) and GM (1.0). AW per SM has strong correlation to WCU (0.86). The strong value of correlation coefficient between metric and component power means it is possible to predict the value of one from another.

B. Components power consumption

Table IV shows components average dynamic power consumption in watts for the HP and LP categories of kernels described in Section III-A. The average dynamic power consumption of HP and LP categories is 80.0 W and 67.2 W respectively.

	RF	EU	WCU	BP	LSU	CP	NOC	MC	GM
HP	11.3	20.2	16.2	3.8	0.6	13.1	2.3	4.2	8.3
LP	4.3	7.0	11.2	3.8	0.9	13.0	4.8	7.8	14.4

TABLE IV: Components dynamic power consumption (W).

The table shows a significant change in components power consumption across the two categories. The EU (25.3%),

WCU (20.3%), and CP (16.3%) are the three most power consuming components for HP category and together consume about 62% of total power. The next most power consuming component is the RF (14.0%). Since these components have higher utilization for kernels with high IPC and hence, these components consume more power. It is interesting to know that the power consumed by the EU (10.4%), WCU (16.6%), and RF (6.4%) is far less for LP category compared to HP category. The largest fraction of power is consumed by the GM (21.4%) in the LP category. The CP and BP power consumption is same in both categories because activation power is consumed in both. The NOC and MC consume more power in LP category because of increased activity of these units. We see that the power distribution is different across the two categories.

C. Low occupancy

In this section we present bottlenecks investigation results for low occupancy category.

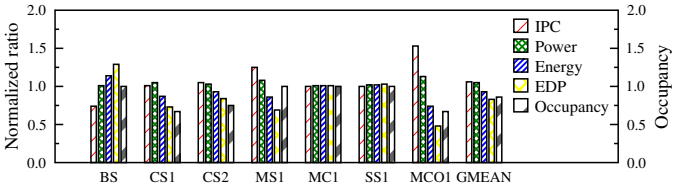
1) Limited by CTA Limit: Table V shows kernels whose occupancy is limited by maximum limit of CTAs. The table shows the kernel, IPC, power, energy consumption, CTA size, and occupancy. The IPC for this category varies from 134.2 to 502.2 and the average IPC is 371.3, which is even less than 37% of the peak IPC. The table also shows that the occupancy varies from 0.33 to 0.67.

Kernel	IPC	Power(W)	Energy(mJ)	CTA size	Occupancy
BS	387.5	167.9	188.1	128	0.67
CS1	339.8	147.3	261.7	64	0.33
CS2	339.8	152.9	260.7	128	0.67
MS1	448.5	154.5	297.8	128	0.67
MC1	502.2	167.1	3.4	128	0.67
SS1	134.2	129.7	2.0	128	0.67
MCO1	446.9	141.2	52.5	64	0.33

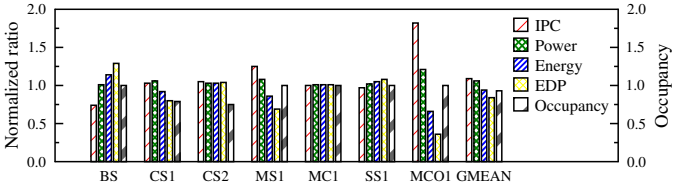
TABLE V: Kernels limited by CTA limit.

Table V shows that the smallest CTA size is 64 threads for the convolution kernel (CS1), and the chroma (MCO1) kernel of H264. These kernels require 24 CTAs per SM to have full occupancy. Thus, we increase the maximum number of CTAs from 8 to 24 in two steps to increase the occupancy. Figure 2 shows IPC, power, energy, EDP normalized to baseline, and occupancy when the CTA limit is increased to 16 and 24. The figure also shows the geometric mean (GMEAN) of all kernels in the category. The average increase in IPC and power is 6% and 5%, respectively, when the CTA limit is increased to 16. The average energy consumption and EDP is decreased by 7% and 17%, respectively. The largest gain is 53% increase in IPC and 26% decrease in energy consumption for the MCO1 kernel. All kernels except BS either gain in IPC or have the same IPC. The reason for the decrease in IPC of BS kernel is that BS is limited by bandwidth utilization (74%). The increase in occupancy adds to the existing pressure on bandwidth, and hence, IPC decreases. Figure 2a shows that the kernels CS1, CS2, and MCO1 still have occupancy <1, and thus, these kernels can gain from further increase in CTA limit. However, CS2 is now limited by shared memory and just increasing the CTA limit further will not help increasing the occupancy of CS2. We call such a kind of bottleneck *second order* bottleneck. Second order bottlenecks may occur after elimination of first order bottlenecks.

Figure 2b shows the IPC, power, energy, EDP, and occupancy when the CTA limit is increased to 24. Only the occupancy of the MCO1 and CS1 kernels increases further because all other kernels either already have full occupancy or show second order bottlenecks after the CTA limit was increased to 16. The average increase in IPC and power is 9% and 6% over the baseline while the average decrease in energy consumption and EDP is 6% and 16%, respectively. The kernel CS1 now also have second order bottleneck of shared memory. The shared memory per SM is increased to 96KB to eliminate the second order bottleneck of CS1, CS2.



(a) Maximum number of CTAs = 16



(b) Maximum number of CTAs = 24

Fig. 2: IPC, power, energy, EDP, and occupancy of kernels limited by CTA limit.

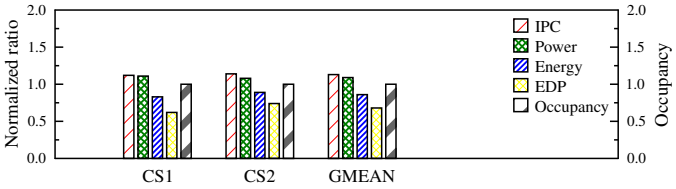


Fig. 3: IPC, power, energy, EDP, and occupancy after second order bottleneck elimination.

Figure 3 shows that the kernels CS1 and CS2 also have full occupancy after the elimination of second order bottleneck. The average increase in IPC of CS1 and CS2 is 13% while average reduction in energy consumption is 14% after the elimination of second order bottleneck. At full occupancy, the average increase in IPC and power for the category is 11% and 7% respectively. The average reduction in energy consumption and EDP is 9% and 23% compared to the baseline. The kernels MC1 and SS1 does not gain in performance even at full occupancy. MC1 gains from increase in memory bandwidth as shown in Section V-E. SS1 has very low coalescing efficiency

(6.7%) and hence just increasing occupancy does not help in increasing the performance.

2) Limited by registers: Table VI shows the kernels whose occupancy is limited by registers. The table shows the kernel, IPC, power, energy consumption, registers used per CTA, and occupancy. The IPC in this category ranges from 26.9 to 517.7 and the average IPC is 343.4 which is 33.5% of the peak IPC. The table shows that the occupancy of these kernels varies from 0.21 to 0.83.

The kernel LC1 has the lowest occupancy and it requires 16K registers per CTA. The LC1 has 320 threads per CTA and requires 4.8 (1536/320) CTAs to reach full occupancy. However, the allocation of threads is done at CTA granularity, thus, the SM can hold a maximum of 4 CTAs in this case. The total number of registers required for 4 CTAs is 64K. However, we only present the results upto 56K registers because at this point either all kernels have full occupancy or a second order bottleneck.

Kernel	IPC	Power(W)	Energy(mJ)	Regs/CTA	Occupancy
BP2	517.7	176.9	30.0	5.5K	0.83
HW	407.6	148.5	3124.4	14.0K	0.67
HS	493.5	154.5	41.9	8.5K	0.50
LC1	271.6	129.8	13283.5	16K	0.21
MUM1	26.9	146.8	714.0	6.0K	0.83

TABLE VI: Kernels limited by registers.

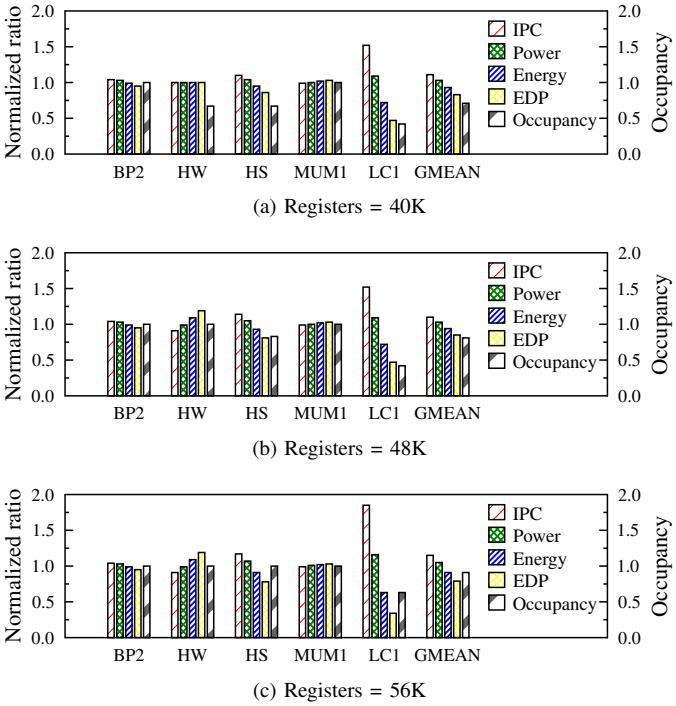


Fig. 4: IPC, power, energy, EDP, and occupancy of kernels limited by registers.

Figure 4 shows IPC, power, energy, EDP and occupancy when the number of registers per SM are increased to 40K, 48K, and 56K. The baseline configuration has 32K registers. Figure 4a shows that the kernels BP2 and MUM1 reach full occupancy after increasing the number of registers to 40K. The largest increase in IPC is 52% for the LC1 kernel, with the corresponding 28% decrease in energy consumption. The average increase in IPC and power is 11% and 3% respectively, while the average decrease in energy consumption and EDP is 7% and 17% respectively. The kernel HW reaches full occupancy when the number of registers are further increased to 48K, but HS and LC1 kernels are still limited by registers and have occupancy less than 1 as shown in Figure 4b. The average increase in IPC and power is 10% and 3% respectively, while the average decrease in energy consumption and EDP is 6% and 15% respectively.

The average increase in IPC and power consumption is 15% and 5% respectively, when the number of registers are further increased to 56K as shown in the Figure 4c. The average decrease in energy consumption and EDP is 9% and 21% respectively. The largest gain is 85% increase in IPC and 37% decrease in energy consumption for the LC1. Since the kernels BP2, MUM1, and HW already have full occupancy at 40K registers. Hence, these kernels do not gain from the increase in registers. The figure shows that all kernels except LC1 have full occupancy. At this point, the occupancy of LC1 is 0.63 and is also limited by a second order bottleneck of shared memory. Hence, we further increase registers size to 64K and also shared memory to 64KB to eliminate the second order bottleneck of LC1. The LC1 reaches its maximum achievable occupancy of 0.83 and continues to gain from increased occupancy. At full occupancy, the average increase in IPC and power for the category is 15% and 5% respectively and the average reduction in energy consumption and EDP is 9% and 21% compared to baseline. The kernels MUM1 and HW does not gain in performance even at full occupancy. MUM1 has high BW utilization (77.2%), low CE (14.9%) and low SU (52.2%) and it gains from increase in memory bandwidth as shown in Section V-E. HW also has low CE (48.4%) and SU (79.6%).

3) Limited by shared memory: There is only one kernel (STO) which is limited by shared memory. STO is used to accelerate a set of hashing functions used in distributed storage systems. The IPC, power, energy, CTA size, shared memory per CTA, registers per CTA, and occupancy is 405.7, 133.8 (W), 51.6 (mJ), 128, 15.9KB, 4.2K, and 0.25, respectively. The IPC is well below the peak IPC and the occupancy is only 25%.

The reason for the low occupancy is that STO is using almost 16KB shared memory per CTA. Since the baseline GPU has 48KB shared memory, no more than 3 CTAs can be allocated simultaneously. Moreover, the CTA size is only 128 which means STO also needs 12 CTAs to achieve full occupancy. Also, STO needs 4.2K registers per CTA. Therefore, at some point, STO will be limited by both CTA and registers limit when the shared memory is increased.

Figure 5 shows the change in performance when the shared memory is increased to 96KB and 144KB, respectively. There is a 49% increase in IPC and a 26% reduction in energy

consumption when the shared memory size is increased to 96KB. The occupancy is doubled to 0.5. There is only a slight increase in occupancy (0.58) when the shared memory is further increased to 144KB because STO is now limited by second order bottleneck of registers. Figure 5 also shows the performance of STO kernel when all second order bottlenecks are eliminated to achieve full occupancy. At full occupancy, the STO kernel gained 85% increase in IPC with just 21% more power consumption. Moreover, we have 35% reduction in energy consumption and 65% less EDP compared to the baseline.

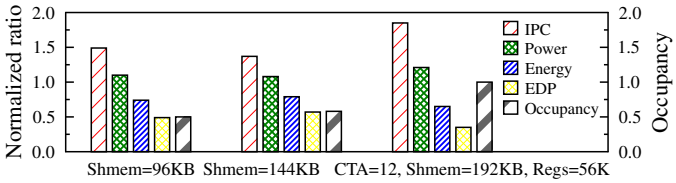


Fig. 5: IPC, power, energy, EDP, and occupancy of the STO.

4) Multiple Bottlenecks: There are two kernels MCO2 and MD which are limited by multiple bottlenecks to begin with. MCO2 is luma kernel of motion compensation part of H264 decoder and it is limited by CTA limit and shared memory. The IPC, power, energy, CTA size, shared memory per CTA, and occupancy of MCO2 is 365.6, 135.6 (W), 183.0 (mj), 64, 6KB, and 0.33, respectively. The kernel has low IPC as well as low occupancy and needs 24 CTAs and 144KB shared memory to have full occupancy. Figure 6 shows that the MCO2 kernel achieves full occupancy when CTA limit is increased to 24 and shared memory size is increased to 144KB. At full occupancy, the IPC is increased by 39% with 8% more power consumption and energy consumption and EDP is decreased by 22% and 44% respectively.

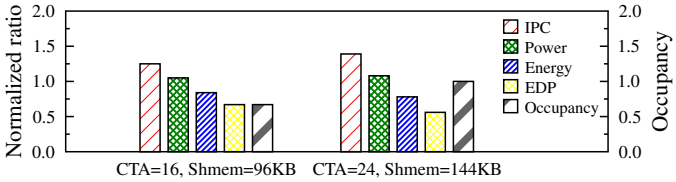


Fig. 6: IPC, power, energy, EDP and occupancy of the MCO2.

MD is used to calculate the physical movements of molecules and atoms and is limited by shared memory and registers. The IPC, power, energy, CTA size, shared memory per CTA, registers per CTA, and occupancy of MD is 142.7, 138.9 (W), 24937.0 (mj), 128, 7.1KB, 4.62K, and 0.5 respectively. We increase the shared memory to 64KB and registers to 48K to increase the occupancy. The occupancy increases to 0.67, but no gain in IPC. MD is now limited by second order bottleneck of CTA limit. We further increase shared memory to 96KB, registers to 56K, and number of CTAs to 12 to have full occupancy. The IPC and power consumption is increased by 2% and 5% respectively. The energy and EDP also increase by 3% and 1% respectively, which shows MD does not gain from increased occupancy. MD does not gain much in IPC from increased occupancy because it has very low coalescing efficiency (13%).

D. Full occupancy

Table VII shows kernels having full occupancy but low performance. The table shows kernel, IPC, power, and energy consumption. The IPC in this category ranges from 8.0 to 468.5. The average IPC is 208.2 which is less than 21% of the peak IPC. Since all kernels in this category have full occupancy, increasing occupancy is not a solution. We analyze if bandwidth utilization (BW), coalescing efficiency (CE), and SIMD utilization (SU) is a bottleneck for low performance.

Kernel	IPC	Power (W)	Energy (mJ)	Kernel	IPC	Power (W)	Energy (mJ)
BT1	432.8	144.7	133.5	SCAN2	286.7	161.1	96.3
BT2	467.0	149.4	123.6	SCAN3	8.0	116.6	10.10
BFS1	21.8	149.9	195.2	SRAD1_1	331.0	163.0	17.1
BFS2	276.5	151.8	10.0	SRAD1_2	370.1	170.3	9.27
CFD1	62.3	144.1	16.7	SRAD1_3	273.1	122.6	9.0
CFD2	184.5	156.6	4.7	SRAD1_4	148.1	148.6	5.9
CFD3	71.0	141.9	5.9	SRAD2_1	304.7	154.0	467.7
FWT1	100.1	154.4	189.7	SRAD2_2	167.0	137.2	452.8
FWT2	264.5	168.4	79.6	MT1	359.9	154.8	3.6
HG3	55.9	125.1	9.4	MT2	46.3	128.2	16.2
KM1	468.5	181.7	741.0	MT3	417.3	156.8	3.5
KM2	11.0	153.2	2216.5	MT4	165.8	134.1	6.6
MUM2	35.62	152.0	681.4	MT5	320.1	152.9	3.5
SP	182.3	141.5	20.7	MT6	144.9	132.5	6.8
VA	171.9	147.4	11.5	MT7	155.8	133.1	6.9
SCAN1	120.4	158.0	57.2	MT8	238.7	151.1	3.1

TABLE VII: Kernels with full occupancy but low performance.

Figure 7 shows BW, CE and SU as a percentage of maximum for the full occupancy kernels. The high BW utilization, low CE, and low SU can severely limit the performance of GPU kernels. Figure 7 shows that kernels CFD1, CFD2, CFD3, FWT1, FWT2, KM1, KM2, MUM2, SP, SCAN1, SCAN2, SRAD1_1, SRAD1_2, SRAD1_4, and VA have high BW utilization and these kernels could be performing low due high bandwidth requirements.

Figure 7 also shows that the kernels HG3 (3%), KM2 (6%), MUM2 (4%), SCAN3 (5%) and MT2 (11%) have very low CE. Also kernels BT1, BT2, BFS1, SCAN1, SCAN2, SRAD1_1, SRAD1_2, SRAD1_3, SRAD2_1, SRAD2_2, MT1, MT3, MT4, MT5, MT6, MT7, and MT8 have less than 100% CE. Low CE could be a reason for their low performance. The low CE results in more than one memory transaction for one memory instruction, resulting in higher pressure on the memory system and larger latencies, and thus, could also limit the performance.

Another factor that could also impact the performance of full occupancy kernels is low SU. The SU is low as a result of branch divergence, likely leading to low performance. For example, a kernel having 50% SU can never have IPC more than 50% of the peak IPC. Figure 7 shows that kernels BFS1 (30.3%) and MUM2 (35.5%) have very low SU and also

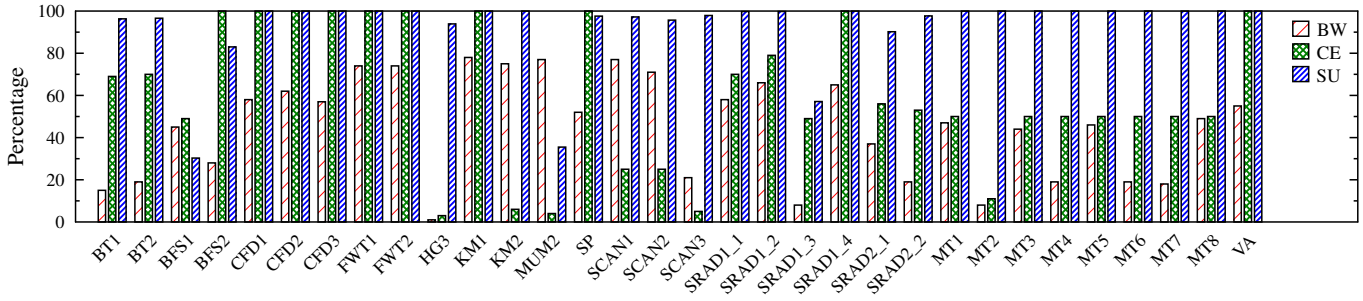
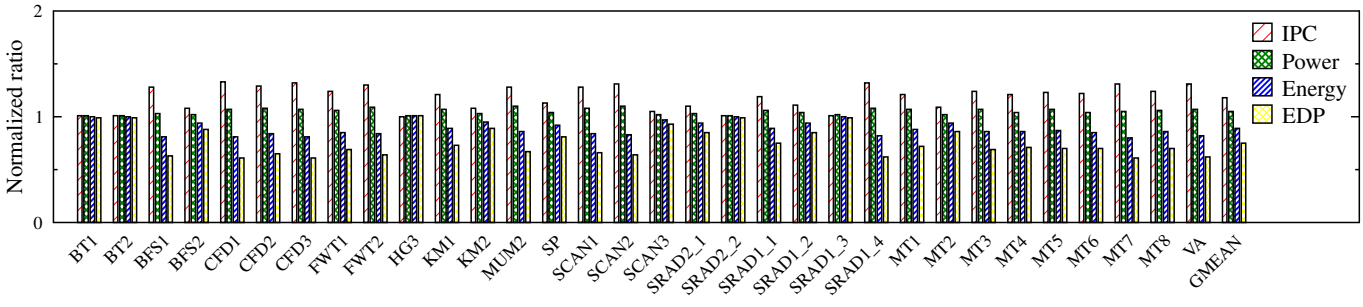
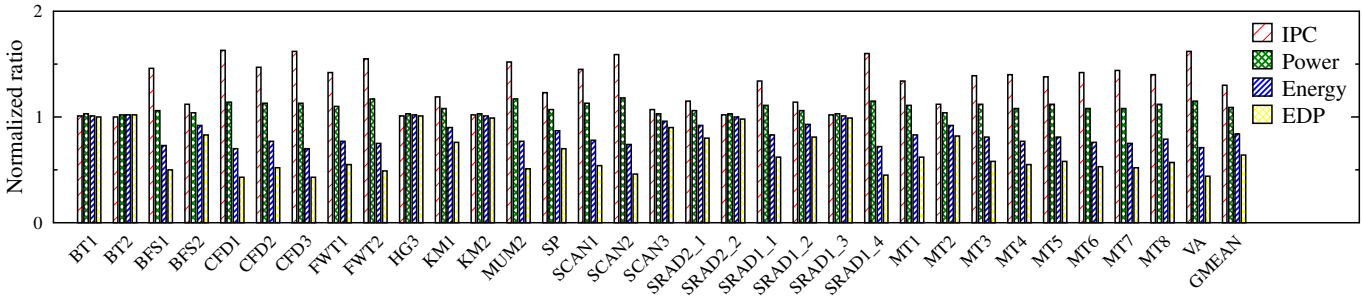


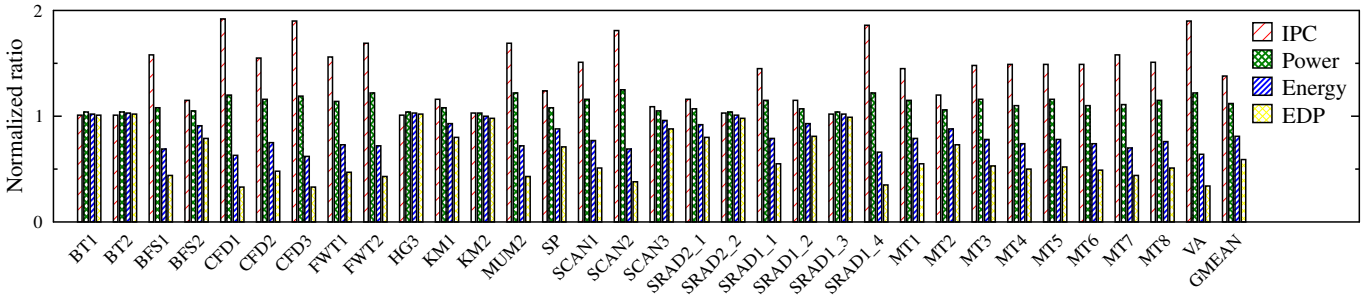
Fig. 7: Bandwidth utilization (BW), Coalescing efficiency (CE), and SIMD utilization (SU) of full occupancy kernels.



(a) Bandwidth = 255.9 GB/s



(b) Bandwidth = 319.4 GB/s



(c) Bandwidth = 384.8 GB/s

Fig. 8: IPC, power, energy, and EDP of full occupancy kernels.

kernels BT1, BT2, BFS2, HG3, SP, SCAN1, SCAN2, SCAN3, SRAD1_3, SRAD2_1, SRAD2_2 have less than 100% SU. The figure also shows that some kernels such as BFS1, MUM2, SRAD1_3 have both low CE and SU and hence, these kernels could have low performance due to the combined effect. All of the matrix transpose kernels have less than 100% CE.

We study the effect of increasing memory bandwidth on full occupancy category. We double the memory bandwidth by doubling the DRAM frequency, incrementing 33.3% at a time and study the change in performance at each increment. The baseline configuration has memory bandwidth of 192.4 GB/s. Thus, we increase memory bandwidth to 255.9 GB/s, 319.4 GB/s, and 384.8 GB/s in three steps.

Figure 8a shows IPC, power, energy, and EDP of full occupancy kernels when the memory bandwidth is increased

to 255.9 GB/s. The average increase in IPC is 18% with 5% more power consumption. Moreover, we see a 11% average reduction in energy consumption and 25% less EDP compared to baseline. Figure 8b shows that kernels gain performance from further increase in memory bandwidth. The average increase in IPC is 30% with only 9% increase in power consumption, while the average decrease in energy consumption and EDP is 16% and 36% compared to baseline. Figure 8c shows that most of the kernels continue to gain from increase in memory bandwidth. The average increase in IPC and power consumption is 38% and 12% respectively, while the average decrease in energy consumption and EDP is 19% and 41% respectively at 384.8 GB/s. The kernels BT1, BT2, HG3, SCAN3, and SRAD1_3 gain very low (avg. 1.3%) from the increase in memory bandwidth because these kernels have low BW utilization (avg. 13%), low CE (avg. 39%), and low SU (avg. 88%).

E. Performance at the combined configuration

In Sections V-C and V-D, we presented bottleneck investigation category-wise. Ideally, we would build a GPU with enough resources so that all kernels achieve optimal performance. Practically, however, it is impossible to build such a GPU due to the area and power demands of the resources combined. Thus, we need to find a design point which captures the benefits for most of the kernels. In this section, we evaluate such a design point. We use the following greedy approach: For each category of kernels, we find an optimal point using maximum reduction in EDP as the criterion. We consider category-wise results up to first order bottlenecks.

Category	Optimal Point
Limited by CTA limit	CTA = 16
Limited by registers	Registers = 56K
Limited by shared memory	Shared memory = 96KB
Multiple bottlenecks	CTA = 24, shared memory = 144KB
Full occupancy	Memory bandwidth = 384.8 GB/s

TABLE VIII: EDP optimal point for each category.

Table VIII shows the EDP optimal point for each category. Then, we combine category EDP optimal points to derive the combined configuration (CTA = 24, registers = 56K, shared memory = 144KB, memory bandwidth = 384.8 GB/s). We choose a larger value of the resource when the resource is common but has different values in two categories to keep the category-wise gains unaffected. For example, the number of CTAs is 16 in CTA limited category and 24 in multiple bottlenecks category and we choose CTAs to be 24 for the combined configuration. This approach will result in a suboptimal solution that can be used to evaluate the effect of all modifications on various categories.

Figure 9 shows the performance of kernels limited by CTA at the combined configuration. The average increase in IPC and power is 31% and 18% respectively, while the average reduction in energy consumption and EDP is 15% and 39% respectively. The increase in performance and energy reduction is higher than the category level and is mainly due

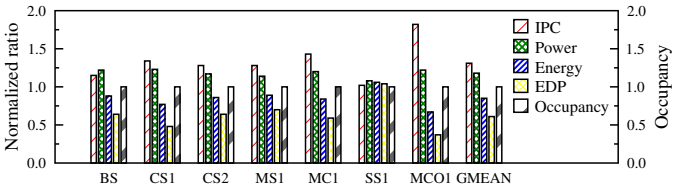


Fig. 9: IPC, power, energy, EDP, and occupancy of kernels limited by CTA limit at the combined configuration.

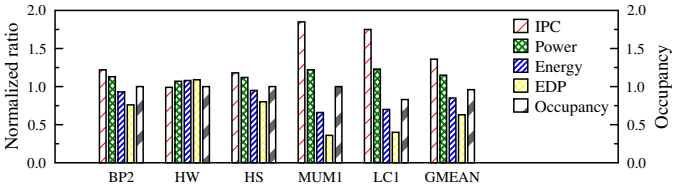


Fig. 10: IPC, power, energy, EDP, and occupancy of kernels limited by registers at the combined configuration.

to the elimination of second order bottlenecks and increased bandwidth.

Figure 10 shows the performance of kernels limited by registers at the combined configuration. The average gain in IPC is 36% which is higher than the average gain at the category level (15%). The average reduction in energy consumption and EDP is 15% and 37% which is also higher than the category-wise gain. The higher gain in performance and energy reduction at the combined configuration shows the registers limited kernels also gain from increased bandwidth.

STO kernel has 56% increase in IPC and 26% decrease in energy consumption compared to baseline at the combined configuration. In the multiple bottlenecks category, MD kernel does not gain in IPC even at the combined configuration because of low CE and MCO2 kernel performance is same as at the category level because it has low BW utilization (3.5%) and hence does not benefit from increased bandwidth at the combined configuration.

The average gain in IPC for the full occupancy kernels at the combined configuration is 38% which is identical to the gain at the category level. This shows that kernels in this category do not gain from other architectural changes, done to increase the occupancy. This is expected as this category already had full occupancy. The average reduction in energy consumption and EDP is 18% and 40% respectively, which is slightly less than the category level. This is caused by the increase in static power due to the increased size of other components.

Table IX shows components dynamic power consumption for LP kernels at the baseline (LP old), combined configuration (LP new) and ratio between them. The component power consumption of RF (48%), EU (35.0%), WCU (13%), LSU (48%), NOC (39%), MC (39%), and GM (57%) increased compared to baseline. This is because the bottlenecks elimination resulted in better utilization of resources which is indicated by higher average IPC (35.5%) compared to baseline. The power consumption of BP and CP remains almost same

because the activation power consumption remains same.

	RF	EU	WCU	BP	LSU	CP	NOC	MC	GM
LP old	4.3	7.0	11.2	3.8	0.9	13.0	4.8	7.8	14.4
LP new	6.4	9.5	12.6	3.7	1.4	13.0	6.7	10.8	22.6
Ratio	1.48	1.35	1.13	0.99	1.48	1.00	1.39	1.39	1.57

TABLE IX: Components dynamic power consumption (W) for LP category kernels at the baseline (LP old), combined configuration (LP new) and their ratio.

VI. Conclusions

We studied the power consumption of GPUs at the component level and correlation between components power consumption and workload metrics. We classified kernels into HP and LP categories. The later has low performance as well as low energy efficiency. The results show a significant change in components power consumption across the two categories.

We also investigated the performance bottlenecks of LP category. The results show that most of the kernels with low occupancy gain in performance and energy efficiency from the increased occupancy. At full occupancy, the average increase in IPC, the average reduction in energy consumption and EDP is 11%, 9% and 23% respectively for CTA limited kernels. The average increase in IPC, the average reduction in energy consumption and EDP is 15%, 9% and 21% respectively for registers limited kernels at full occupancy. The results show that high occupancy is an important factor for both high performance and energy efficiency.

We further show that full occupancy kernels have low performance either due to high BW utilization or low CE or low SU. The full occupancy kernels on an average have 38% increase in IPC, 19% decrease in energy consumption and 41% reduction in EDP at double bandwidth. However, we also found that only few kernels (9 out of 47) could achieve IPC greater than 50.5% of the peak IPC. We conclude that increased occupancy and bandwidth does help in increasing the performance and reducing the energy consumption, but it is alone not enough to achieve the maximum performance for most of the kernels. We also show that many kernels in full occupancy category are severely limited by low CE.

In the future work, we would investigate the architectural implications of increasing the CTA limit, registers, shared memory in detail. Studying the bottlenecks that low CE creates at different levels in memory hierarchy would also be very interesting and any opportunities for micro-architectural changes that could benefit the kernels suffering from very low CE should be explored.

VII. Acknowledgements

This project receives funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the LPGPU Project (www.lpgpu.org), grant agreement n° 288653.

References

[1] S. Huang, S. Xiao, and W. Feng, "On the Energy Efficiency of Graphics Processing Units for Scientific Computing," in *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, ser. IPDPS, 2009.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark Silicon and the End of Multicore Scaling," in *Proceedings of the 38th annual international symposium on Computer architecture*, ser. ISCA, 2011.

[3] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs and the Future of Parallel Computing," *IEEE Micro*, vol. 31, sept–oct. 2011.

[4] X. Ma, M. Dong, L. Zhong, and Z. Deng, "Statistical Power Consumption Analysis and Modeling for GPU-based Computing," in *Proceedings of the Workshop on Power Aware Computing and Systems, HotPower*, 2009.

[5] M. Gebhart, D. R. Johnson, D. Tarjan, S. W. Keckler, W. J. Dally, E. Lindholm, and K. Skadron, "Energy-Efficient Mechanisms for Managing Thread Context in Throughput Processors," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ser. ISCA, 2011.

[6] J. Lucas, S. Lal, M. Andersch, M. A. Mesa, and B. Juurlink, "Why a Single Chip Causes Massive Power Bills - GPUSimPow: A GPGPU Power Simulator," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, ser. ISPASS, 2013.

[7] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "GPUWattch: Enabling Energy Optimizations in GPGPUs," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ser. ISCA, 2013.

[8] A. Kerr, G. Diamos, and S. Yalamanchili, "A Characterization and Analysis of PTX Kernels," in *IEEE International Symposium on Workload Characterization*, ser. IISWC, 2009.

[9] S. Che, J. Sheaffer, M. Boyer, L. Szafaryn, L. Wang, and K. Skadron, "A Characterization of the Rodinia Benchmark Suite with Comparison to Contemporary CMP Workloads," in *IEEE International Symposium on Workload Characterization*, ser. IISWC, 2010.

[10] N. Goswami, R. Shankar, M. Joshi, and T. Li, "Exploring GPGPU Workloads: Characterization Methodology, Analysis and Microarchitecture Evaluation Implications," in *IEEE International Symposium on Workload Characterization*, ser. IISWC, 2010.

[11] M. Burtscher, R. Nasre, and K. Pingali, "A Quantitative Study of Irregular Programs on GPUs," in *IEEE International Symposium on Workload Characterization*, ser. IISWC, 2012.

[12] E. Blem, M. Sinclair, and K. Sankaralingam, "Challenge Benchmarks that Must be Conquered to Sustain the GPU Revolution," in *Proceedings of the 4th Workshop on Emerging Applications for Manycore Architecture*, 2011.

[13] A. Bakhoda, G. Yuan, W. Fung, H. Wong, and T. Aamodt, "Analyzing CUDA Workloads Using a Detailed GPU Simulator," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS*, 2009.

[14] V. Volkov, "Better Performance at Lower Occupancy," in *GPU Technology Conference*, 2010.

[15] S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A Benchmark Suite for Heterogeneous Computing," in *Proceedings of the IEEE International Symposium on Workload Characterization*, 2009.

[16] NVIDIA, "CUDA: Compute Unified Device Architecture," 2007, http://developer.nvidia.com/object/gpucomputing.html.

[17] S. Al-Kiswany, A. Gharaibeh, E. Santos-Neto, G. Yuan, and M. Ripeanu, "StoreGPU: Exploiting Graphics Processing Units to Accelerate Distributed Storage Systems," in *Proc. of the 17th International Symposium on High Performance Distributed Computing*, ser. HPDC, 2008.