Ashraf E. Suyyagh

GPGPU Workload Analysis Based on CUDA Kernels

Outline

- Design Options Simulated
- Simulator
- Workload
- Results
- Conclusions

Design Options

- A base configuration similar to contemporary high end GPU designs was chosen.
- Design parameters either were related to architectural design aspects, ratio of processors/warp size, No. of registers and size of shared memory per SM.
- Other parameters related to No. of threads per thread block, exploring Thread Block coarse grained level parallelism.

Hardware Simulated	Basic Configuration	Different Configurations Simulated
No. of Streaming Multiprocessors	28	-
No. of processors per SM	32	8/16/32
No. of threads in thread block	1024	512/1024/1536/2048
No. of registers / SM	16384	4096/8192/16384/24576/32768
Shared memory size (bytes)/ SM	16384	16384/24576/32768
No. of concurrent thread Blocks	8	4/8/12/16

Simulator

- Few GPU Simulators available: Barra, Ocelot, GPGPU Sim
- GPGPU Sim provides detailed statistical results and allows for much wider range of design and simulation options
- Offers Functional and Performance Simulation Options

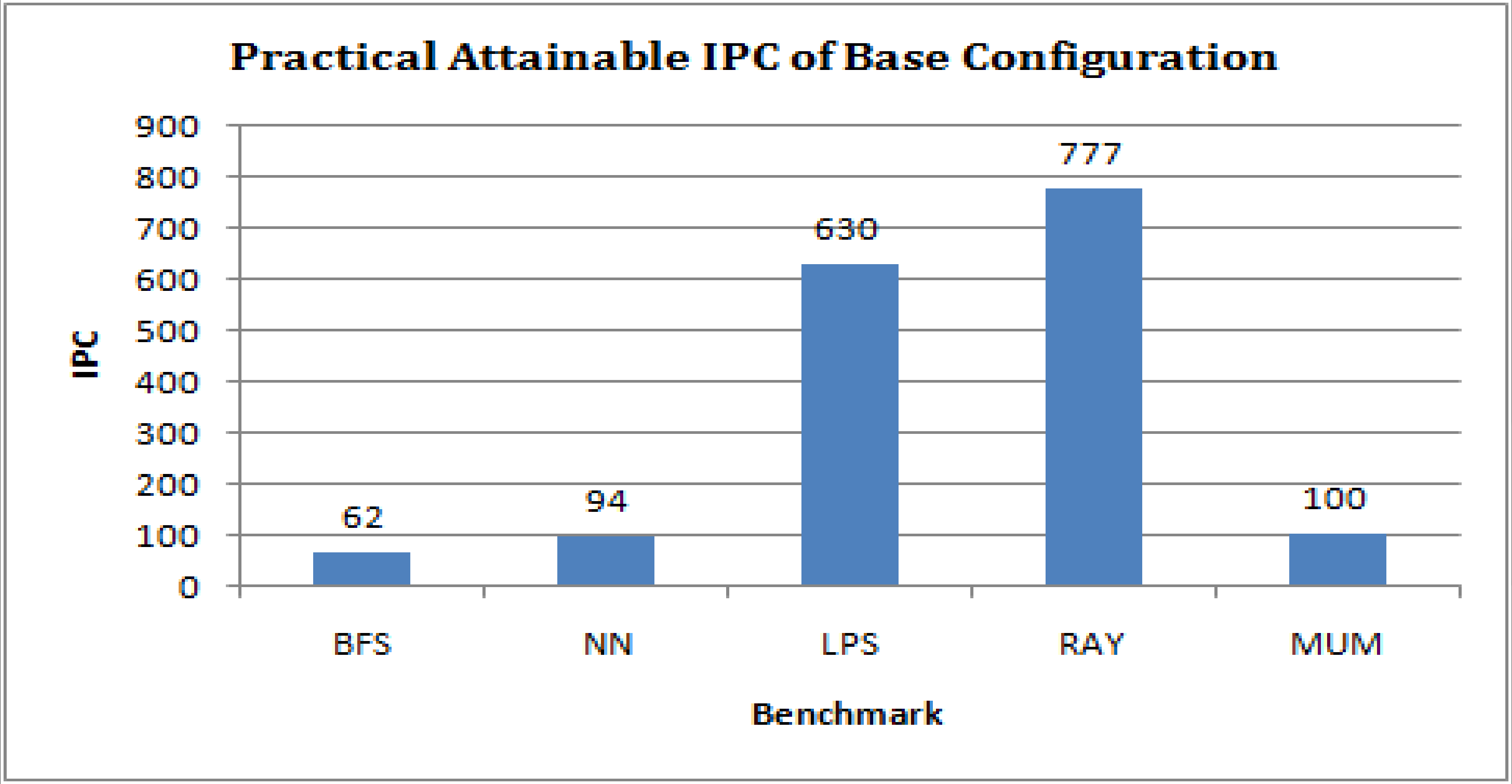
Workload

- No official benchmark suite has yet been developed for general purpose computing on GPUs
- Researchers use some of the highly complex kernels provided by the NVIDIA CUDA SDK
- Some compile their own sets of general purpose applications
- This simulation used a subset of the set used by Bakhoda et al in their simulation work!

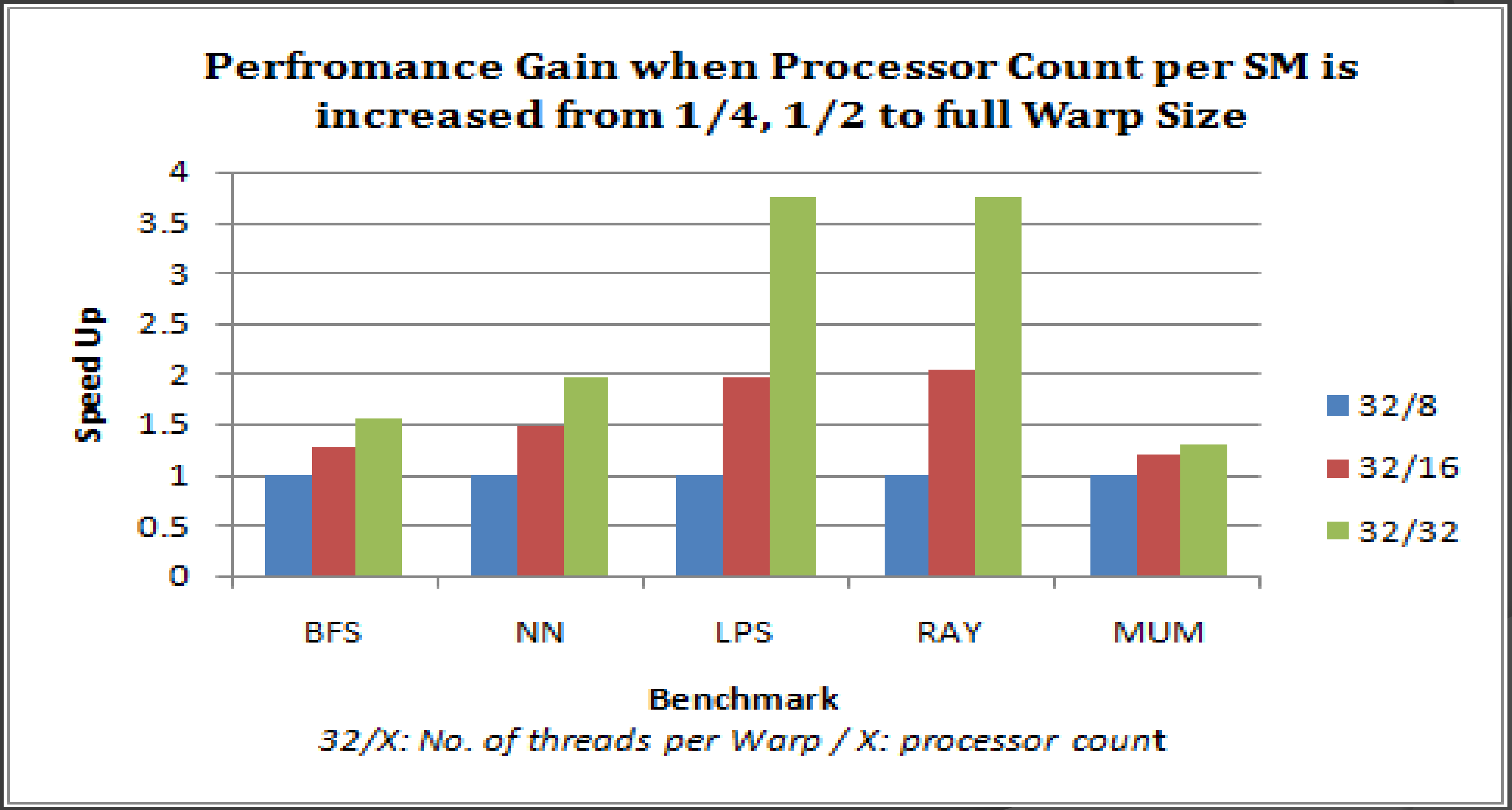
Workload II - Properties

Benchmark	Grid Dimension	Thread Block Dimensions	Concurrent Thread Blocks/SM	Total Threads	Shared Memory	Constant Memory	Texture Memory	Barriers
BFS	128,1,1	512,1,1	4	65563	Y	N	N	N
LPS	4,25,1		6	12800	Y	N	N	Y
NN	6,28,1 50,28,1 100,28,1 10,28,1	13,13,1 5,5,1 1,1,1 1,1,1	5 8 8 8	28392 35000 2800 280	N	N	N	N N N N
MUM	782,1,1	64,1,1	3	50000	N	N	2D	N
RAY	16,32,1	16,8,1	3	65563	N	Y	N	Y

Results – Base Configuration

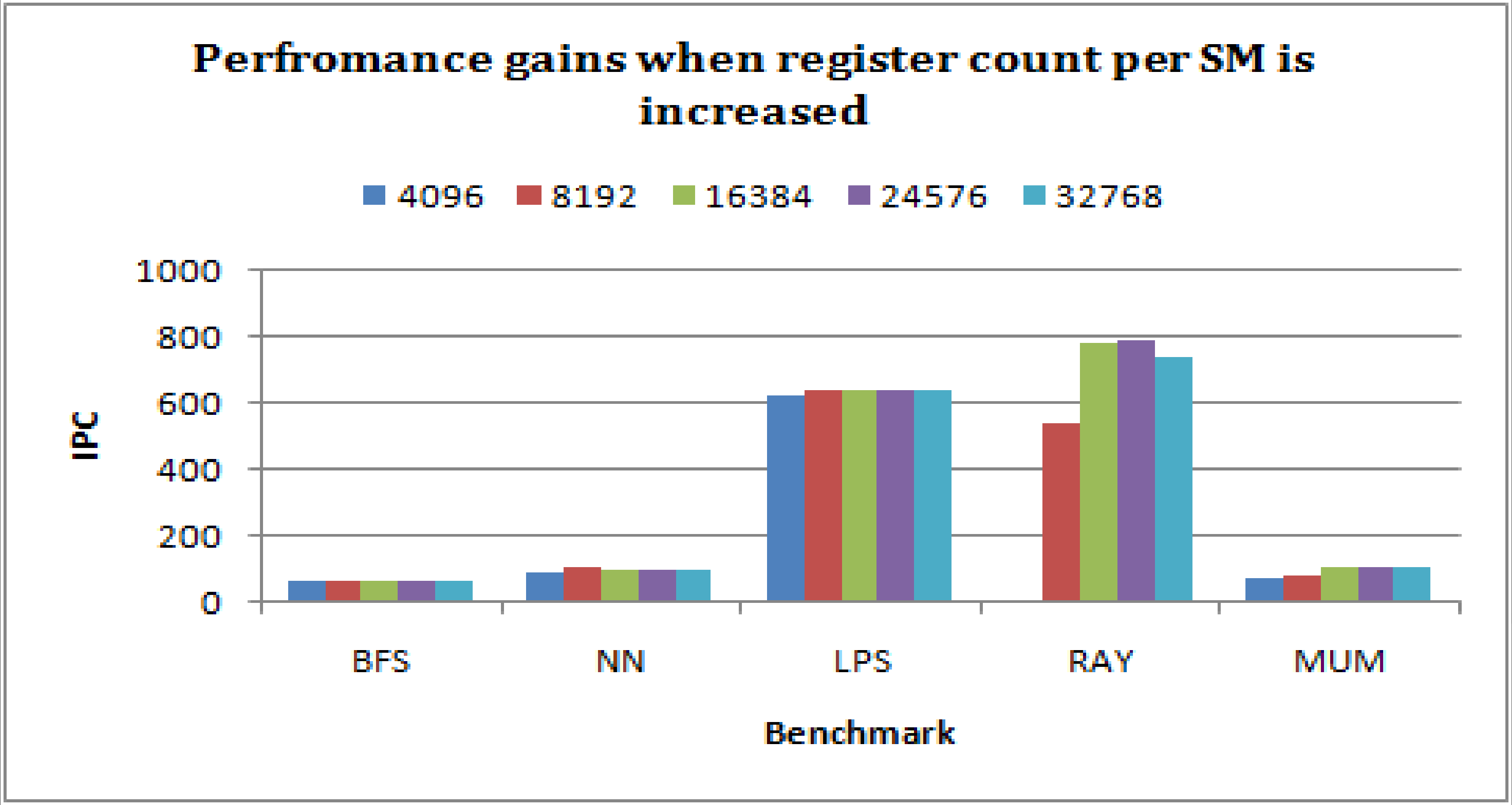


Results II



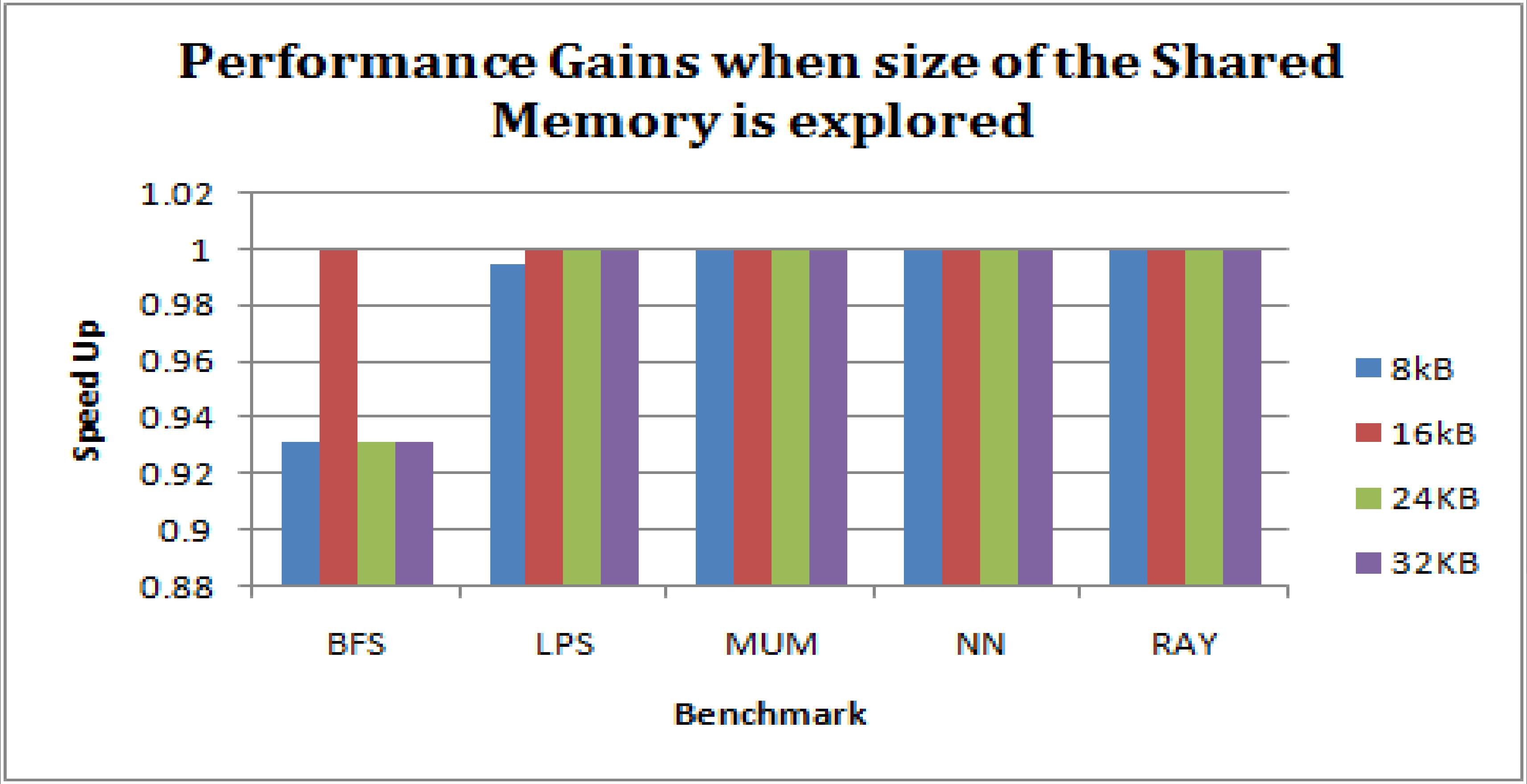
Perfromance Gain when Processor Count per SM is increased from 1/4, 1/2 to full Warp Size
Speed Up
4
3.5
3
2.5
2
1.5
1
0.5
0
BFS
NN
LPS
RAY
MUM
32/8
32/16
32/32
Benchmark
32/X: No. of threads per Warp / X: processor count

Results III

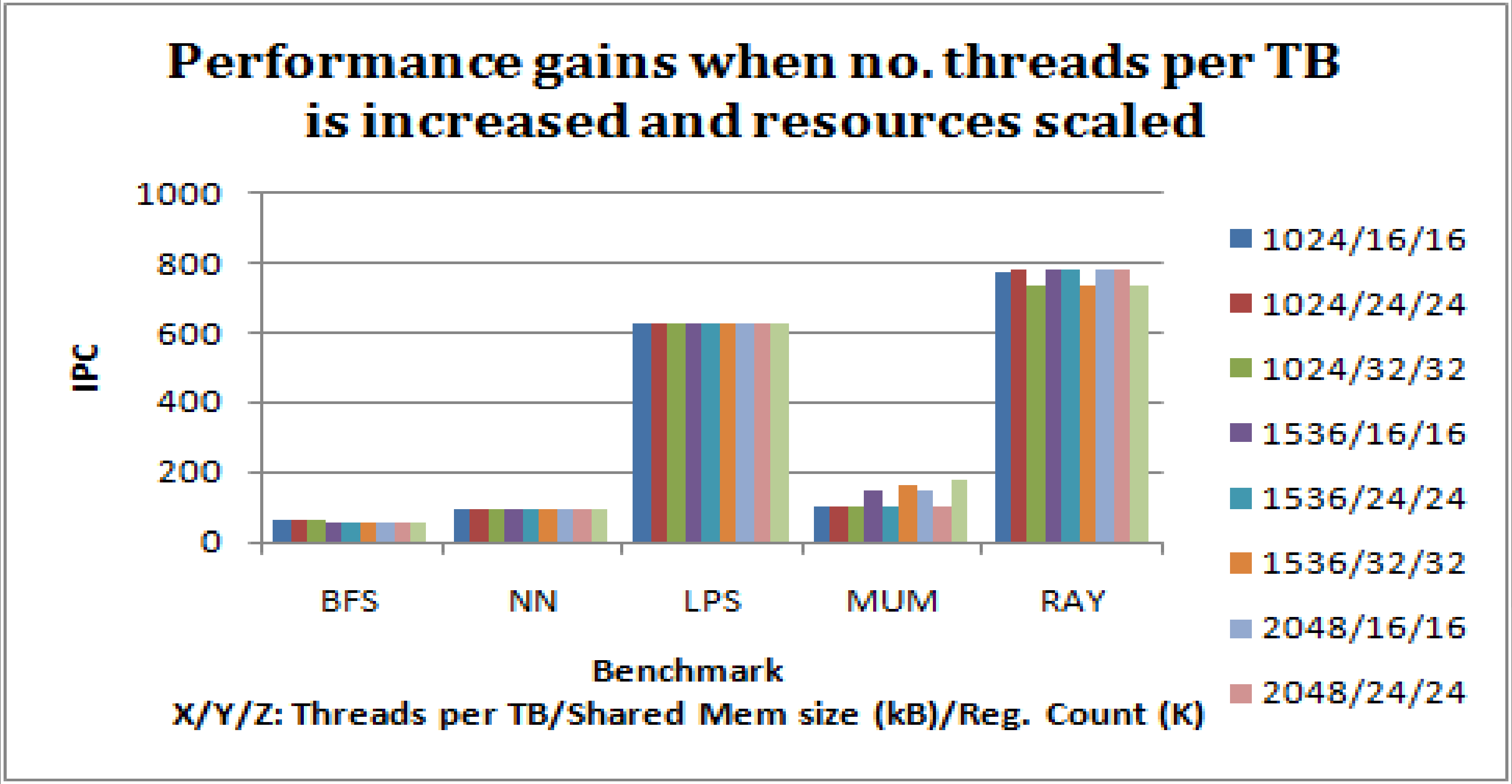


Perfromance gains when register count per SM is increased
4096
8192
16384
24576
32768
1000
800
600
400
200
0
IPC
BFS
NN
LPS
RAY
MUM
Benchmark

Results IV

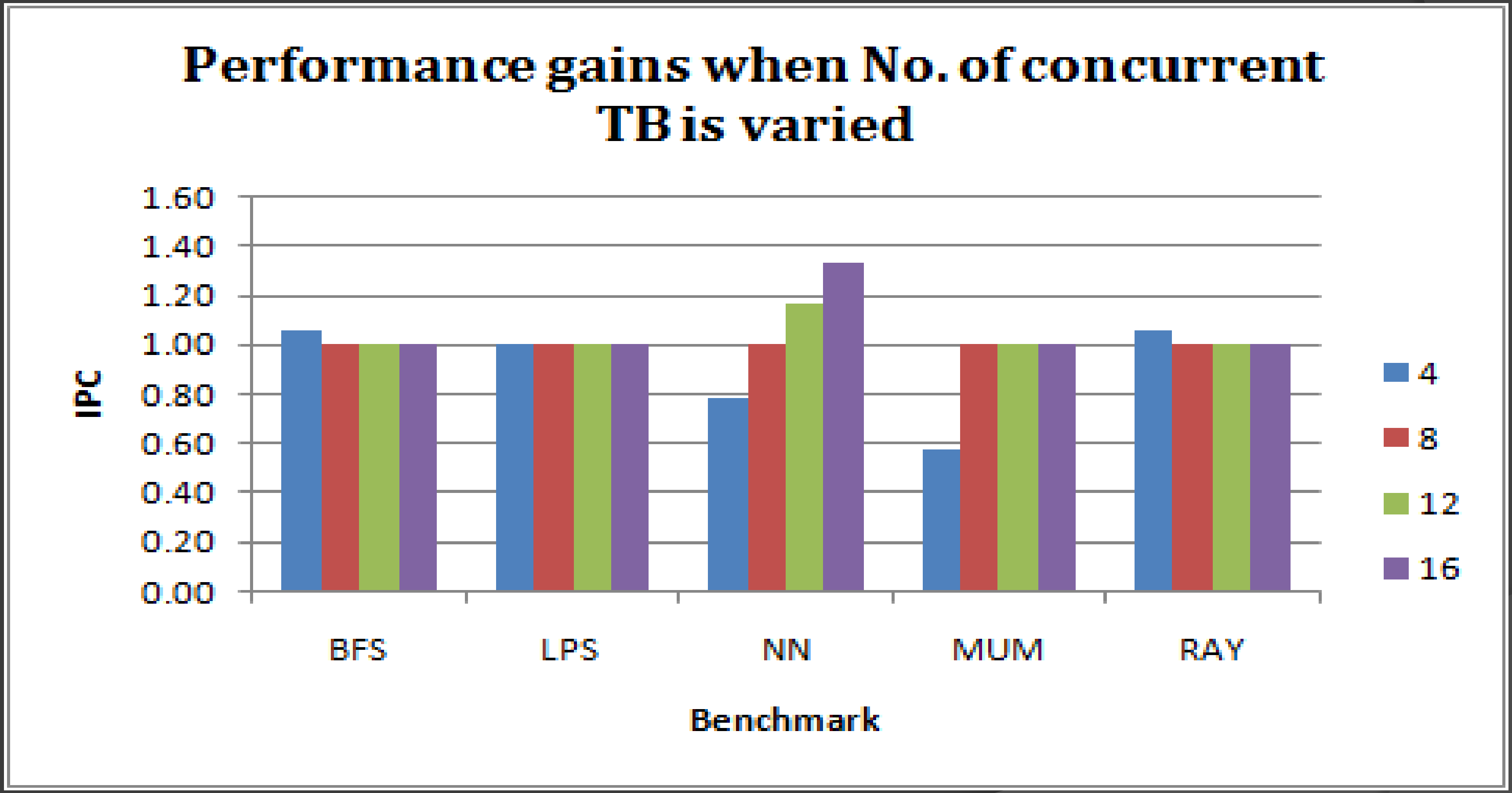

Performance Gains when size of the Shared Memory is explored
Speed Up
1.02
1
0.98
0.96
0.94
0.92
0.9
0.88
BFS
LPS
MUM
NN
RAY
Benchmark
8kB
16kB
24KB
32KB

Results V



Performance gains when no. threads per TB is increased and resources scaled
1000
800
600
400
200
0
IPC
BFS
NN
LPS
MUM
RAY
Benchmark
X/Y/Z: Threads per TB/Shared Mem size (kB)/Reg. Count (K)
1024/16/16
1024/24/24
1024/32/32
1536/16/16
1536/24/24
1536/32/32
2048/16/16
2048/24/24

Results VI


Performance gains when No. of concurrent TB is varied
IPC
1.60
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
BFS
LPS
NN
MUM
RAY
Benchmark
4
8
12
16

Conclusions

- Increasing No. of shader cores doesn't necessarily scale performance linearly → No completely parallel programs, branch divergence
- Increasing shared memory size and register count doesn't scale performance when it surpasses the amount needed by applications
- Increasing No. of threads though expected to enhance performance – limited by global memory access and interconnect congestion
- Thread Block Coarse grained level of parallelism is limited by the amount of independent thread blocks in the kernel!

Thank You!

Ashraf Suyyagh
a.suyyagh@ju.edu.jo