Multi Core Architecture

Assignment 1

GPU and CUDA Intro

Group members,
Shriraam Mohan <shriraam.mohan@mailbox.tu-berlin.de>
Tamilselvan Shanmugam <tamilselvan@mailbox.tu-berlin.de>

Due date: 19 Jan 2015, Monday

Implementation:

julia.c has been ported to cuda. Calculating image pixes are carried out by individual threads. Totally 256 x 256 = 65536 threads are needed to draw the julia image. We have tested kernel with 1, 16, 32, 256 1D block sizes and 4x4, 8x8, 8x32, 32x8, 16x16 2D block sizes.

Execution time:

1D Block size	gpu_sim_cycle	gpu_sim_insn	gpu_ipc	Stall	W0_Idle	W0_Score board	Total waiting time
1	2965890	52463967	17.6891	176846	114219	32754417	33045482
16	353828	52463967	148.2753	21500	112937	3713443	3847880
32	233178	52463967	224.9954	14027	142585	2395960	2552572
256	196587	52463967	266.8741	594700	497261	224283	1316244

2D Block size	gpu_sim_cycle	gpu_sim_insn	gpu_ipc	Stall	W0_Idle	W0_Score board	Total waiting time
4x4	323937	53578079	165.3966	18664	116632	3390962	3526258
8x8	158319	53578079	338.4185	289423	100845	610534	1000802
8x32	172417	53578079	310.7471	505735	328079	245084	1078898
32x8	198648	53578079	269.7137	618266	395605	219351	1233222
16x16	179859	53578079	297.8893	552175	334745	225200	1112120

(Total waiting time = Stall + W0_Idle+W0_Score_board)

From the above table it's clearly evident that the execution time greatly depends on the total number of wait cycles. A thread may wait for memory operations, may be inserted in to pipeline and still not got the processor are some examples of such wait states. As it goes high, execution time also grows. Tuning application to have minimum wait time is a key to improve execution time.

Performance:

As our SIMD width is 32, every block is packed in to multiples of 32 threads and sent to streaming multiprocessor unit. When a block is initialized with only one thread, kernel assembles it in to one wrap with only one thread. Whereas, a wrap is capable of executing 32 threads. It means that the 31 SIMD units will be idle all the time and only 1 will be doing productive work. It drastically reduces the SIMD efficiency. This applies to all the cases. Block size of 16 also failed to exploit 16 SIMD units, resulting poor performance. Block size of 32 and above can aim to utilize all the available units, thus producing maximum efficiency.

The same reasoning applies to 2D blocks. Size 4x4 has 16 threads in a block. It lacks another 16 threads to gain SIMD efficiency. All other block sizes have ample amount of threads to use all the SIMD units and achieved reasonably high performance.

1D Block sizes and SIMD Efficiency:

Wn	1
W1	55782700
W2	0
W3	0
W4	0
W5	0
W6	0
W7	0
W8	0
W9	0
W10	0
W11	0
W12	0
W13	0
W14	0
W15	0
W16	0
W17	0
W18	0
W19	0
W20	0
W21	0
W22	0
W23	0
W24	0
W25	0
W26	0
W27	0
W28	0
W29	0
W30	0
W31	0
W32	0
Tot	55782700
Efficiency	**3.13%**

Wn	16
W1	1448772
W2	709748
W3	416580
W4	302180
W5	229012
W6	186304
W7	153888
W8	123984
W9	108368
W10	92624
W11	83680
W12	80336
W13	61248
W14	56400
W15	56592
W16	2493952
W17	0
W18	0
W19	0
W20	0
W21	0
W22	0
W23	0
W24	0
W25	0
W26	0
W27	0
W28	0
W29	0
W30	0
W31	0
W32	0
Tot	6603668
Efficiency	**26.40%**

Wn	32
W1	934568
W2	462518
W3	292862
W4	198174
W5	154462
W6	127232
W7	100688
W8	89872
W9	75392
W10	71232
W11	64816
W12	59872
W13	47008
W14	39616
W15	38016
W16	34624
W17	32736
W18	29936
W19	34912
W20	26208
W21	26816
W22	27920
W23	27312
W24	27328
W25	27648
W26	24640
W27	22160
W28	23824
W29	17904
W30	16752
W31	18592
W32	1101264
Tot	4276904
Efficiency	**40.76%**

Wn	256
W1	934568
W2	462518
W3	292862
W4	198174
W5	154462
W6	127232
W7	100688
W8	89872
W9	75392
W10	71232
W11	64816
W12	59872
W13	47008
W14	39616
W15	38016
W16	34624
W17	32736
W18	29936
W19	34912
W20	26208
W21	26816
W22	27920
W23	27312
W24	27328
W25	27648
W26	24640
W27	22160
W28	23824
W29	17904
W30	16752
W31	18592
W32	1101264
Tot	4276904
Efficiency	**40.76%**

2D Block sizes and Efficiency:

Wn	4x4
W1	1118294
W2	554096
W3	329598
W4	222730
W5	163364
W6	123664
W7	100576
W8	92698
W9	75008
W10	61056
W11	62176
W12	61872
W13	54768
W14	55568
W15	68512
W16	2784784
W17	0
W18	0
W19	0
W20	0
W21	0
W22	0
W23	0
W24	0
W25	0
W26	0
W27	0
W28	0
W29	0
W30	0
W31	0
W32	0
Tot	5928764
Efficiency	**29.99%**

Wn	8x8
W1	643124
W2	330156
W3	219420
W4	145082
W5	107962
W6	95098
W7	77194
W8	59322
W9	54784
W10	45104
W11	39776
W12	35664
W13	32368
W14	30288
W15	27088
W16	28592
W17	22688
W18	20160
W19	19600
W20	19760
W21	17280
W22	17152
W23	16752
W24	17552
W25	15248
W26	14112
W27	13792
W28	15632
W29	17696
W30	19120
W31	22080
W32	1322464
Tot	3562110
Efficiency	**49.91%**

Wn	8x32
W1	643124
W2	330156
W3	219420
W4	145082
W5	107962
W6	95098
W7	77194
W8	59322
W9	54784
W10	45104
W11	39776
W12	35664
W13	32368
W14	30288
W15	27088
W16	28592
W17	22688
W18	20160
W19	19600
W20	19760
W21	17280
W22	17152
W23	16752
W24	17552
W25	15248
W26	14112
W27	13792
W28	15632
W29	17696
W30	19120
W31	22080
W32	1322464
Tot	3562110
Efficiency	**49.91%**

Wn	32x8
W1	934568
W2	462518
W3	292862
W4	198174
W5	154462
W6	127232
W7	100688
W8	89872
W9	75392
W10	71232
W11	64816
W12	59872
W13	47008
W14	39616
W15	38016
W16	34624
W17	32736
W18	29936
W19	34912
W20	26208
W21	26816
W22	27920
W23	27312
W24	27328
W25	27648
W26	24640
W27	22160
W28	23824
W29	17904
W30	16752
W31	18592
W32	1136080
Tot	4311720
Efficiency	**41.24%**

Wn	16x16
W1	736924
W2	390608
W3	236500
W4	170576
W5	125476
W6	108432
W7	83354
W8	69392
W9	64074
W10	56288
W11	48960
W12	45440
W13	40592
W14	38960
W15	32736
W16	32480
W17	27936
W18	25296
W19	21840
W20	23696
W21	21728
W22	23008
W23	18608
W24	21488
W25	15728
W26	15856
W27	14944
W28	14896
W29	15072
W30	18032
W31	17104
W32	1262608
Tot	3838632
Efficiency	**46.32%**

Calculating optimal Block Size is tedious task. Many factors like number of registers available per block and thread, allocated shared memory, threads per block come into play to determine occupancy of multiprocessor. http://bit.ly/1xygvjS gives nice assessment of block size, number of threads and corresponding performance.