In the name of God

Multicore Programming Homework 5

Amir M Pirhosseinloo

9531014

1. Peak Bandwidth: 64 * 2 * 795 / 8 MHz = 12.42 GHz

2. kernels:

a. reduce0 (interleaved addressing with divergent branching): در این کرنل، تعداد نخ های فعال به صورت نمایی با افزایش عمق درخت محاسبات کاهش می یابد. در این کرنل branch diversion زیاد است.

b. reduce1 (interleaved addressing with bank conflict): در این کرنل مشکل branch diversion حل شده است اما مشکل حافظه مشترک به وجود آمده است.

c. reduce2 (sequential addressing): در این روش همانند reduce0، با افزایش عمق درخت محاسبات تعداد نخ های فعال کاهش می یابد اما مزیت آن این است که درخواست ها به خانه های مجاور coalesce (تلفیق) شده است و زمان لود کردن داده ها به داخل حافظه مشترک کمتر شده است.

d. reduce3 (first add during global add): در روش reduce2 در گام اول محاسبات تنها نصف نخ های هر بلاک فعال هستند که یک مشکل محسوب می شود. در این روش این مشکل به صورت زیر حل شده است: نخ ها در همان موقع که در حال بارگذاری داده در خانه های حافظه مشترک هستند، خانه معادل در بلاک بعدی را هم با خانه فعلی جمع کرده سپس بارگذاری می کنند که باعث می شود تعداد بلاک های مورد نیاز نصف شود و همچنین در گام اول همه نخ ها فعال هستند.

e. reduce4 (unroll last warp): در روش های قبلی در گام های آخر تعداد نخ های فعال به شدت کاهش می یابد و sync کردن آن ها overhead زیادی تولید می کند (نخ های فعال آخر در یک warp هستند و خود به خود sync هستند). در روش جاری، این مشکل این گونه حل شده است که نخ های آخر را از حلقه خارج کرده و کار آن ها را بدون استفاده از __syncthreads انجام می دهیم.

3. results

kernel	Total Time (134217728 ints) (ms)	Computation Time (ms)	bandwidth	step speedup	cumulative speedup
reduce0	666	190	24.02 MB/s	-	-
reduce1	694	102	44.67 MB/s	1.86	1.86
reduce2	674	87	52.36 MB/s	1.17	2.18
reduce3	563	45	101.36 MB/s	1.93	4.22
reduce4	611	20	228.19 MB/s	2.25	9.5

4. serial version (reduction.cu) execution time with 134217728 elements in vector: 3341 ms
   best execution time on GPU = 20 ms
   speed up = 167.05 (considering just execution time)
   speed up = 5.46 (considering execution time + data exchange between CPU and GPU)
   First comparison is more fair than second.
   more data -> more time to send data from CPU to GPU and more time to send it back -> significant change in speed up while considering total time for computing speed up (reduces speed up).
5. All kernels start computation with 524288 blocks and 256 threads per block except reduce_3 and reduce_4 which start with 256144 block.