



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

تمرین اول سیستم‌های موازی

نگارش:

مهسا قادران

۴۰۰۲۰۱۰۴۸

۸ فروردین ۱۴۰۱

مشخصات کارت گرافیک

```
CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "GeForce 940MX"
  CUDA Driver Version / Runtime Version      10.2 / 10.2
  CUDA Capability Major/Minor version number: 5.0
  Total amount of global memory:              2004 MBytes (2101870592 bytes)
  ( 3) Multiprocessors, (128) CUDA Cores/MP:  384 CUDA Cores
  GPU Max Clock rate:                         1189 MHz (1.19 GHz)
  Memory Clock rate:                          2000 Mhz
  Memory Bus Width:                           64-bit
  L2 Cache Size:                              1048576 bytes
  Maximum Texture Dimension Size (x,y,z)      1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers 1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers 2D=(16384, 16384), 2048 layers
  Total amount of constant memory:             65536 bytes
  Total amount of shared memory per block:     49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                   32
  Maximum number of threads per multiprocessor: 2048
  Maximum number of threads per block:         1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                       2147483647 bytes
  Texture alignment:                           512 bytes
  Concurrent copy and kernel execution:        Yes with 1 copy engine(s)
  Run time limit on kernels:                   Yes
  Integrated GPU sharing Host Memory:           No
  Support host page-locked memory mapping:      Yes
  Alignment requirement for Surfaces:           Yes
  Device has ECC support:                      Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Compute Preemption:           No
  Supports Cooperative Kernel Launch:          No
  Supports MultiDevice Co-op Kernel Launch:     No
  Device PCI Domain ID / Bus ID / location ID: 0 / 1 / 0
  Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
```

سوال اول: مرتب‌سازی مرتبه‌ای

در این سوال یک آرایه جهت تشکیل آرایه مرتب شده به پردازنده گرافیکی تخصیص می‌دهیم. همه رشته‌ها کل آرایه را پیمایش میکنند و هرباری که خانه‌ای با مقدار کوچک‌تر از خودشان ببینند مقدار یک متغیر شمارنده محلی را افزایش می‌دهند. مقدار نهایی شمارنده برابر اندیس آن خانه از آرایه

در آرایه نهایی مرتب شده است.

در الگوریتم ذکر شده در صورتی که در آرایه ورودی چند خانه با مقدارهای مشابه وجود داشته باشند اندیس همه‌شان باهم برابر خواهد بود. برای جلوگیری از رخداد این اتفاق، هربار که رشته خانه‌ای از آرایه هم مقدار با خانه معادل خودش ببیند در صورتی که شناسه‌ای کوچک تر از خودش داشته باشد مقدار شمارنده را افزایش می‌دهد. در چنین حالتی، خانه‌های با مقادیر مشابه به ترتیب اندیس‌ها مرتب می‌شوند و اندیس‌های متمایزی در آرایه خروجی خواهند داشت.

سوال دوم: مرتب‌سازی جابه‌جایی زوج و فرد

در این سوال جهت مرتب‌سازی یک آرایه با تعداد N اندیس، به تعداد $N/2$ رشته نیاز داریم. الگوریتم به این صورت عمل میکند که در راندهای زوج رشته‌ها سراغ اندیس‌های زوج از آرایه مشترک می‌روند در حالی که در راندهای فرد سراغ اندیس‌های فرد از آرایه می‌روند. و در هر دور از اجرای هر رشته مقدار خود را با خانه سمت راست خود مقایسه کرده و در صورتی که مقدار سمت راست کوچک‌تر از مقدار خودش باشد مقدار خانه‌ها را جابه‌جا می‌کند. این الگوریتم پس از N دور خانه‌های آرایه را مرتب می‌کند. و مستقل از اندازه آرایه است.

سوال سوم: جمع پیش‌وندی بهبودیافته

این الگوریتم از دو فاز کلی تشکیل شده است.

فاز صعودی: در این فاز، تعداد thread ها برابر نصف آرایه ورودی می‌باشد. سپس تعداد رشته‌ها در هر راند نصف میشود. تا جایی که نهایتاً یک رشته داریم. در این فاز ترتیب جمع اعداد این صورت است که خانه‌هایی با که اندیسشان مضرب $2^{round} + 1$ است با $2^{round/2}$ خانه قبل خود جمع می‌شوند.

فاز نزولی: در این فاز تعداد رشته‌ها از یک شروع شده و تا $2^k - 1$ پیش می‌رود. و در هر دوره فاصله اندیس‌هایی که با همدیگر جمع زده می‌شوند نصف میشود. الگوریتم هنگامی پایان می‌پذیرد که نهایتاً هر خانه با اندیس زوج با مقدار خانه قبلی خود جمع شود.

در این الگوریتم همان طور که اشاره شد تنها با نصف اندازه آرایه به عبارتی $N/2$ رشته قابل پیاده‌سازی است. جهت پیاده‌سازی الگوریتم آرایه مورد نظر در قسمت حافظه مشترک میان رشته‌ها قرار می‌گیرد. هر فاز از الگوریتم در قالب یک for loop پیاده‌سازی شده است و رشته‌ها قبل از رفتن به راند بعدی با استفاده از دستور synctreads با یکدیگر همگام می‌شوند. نکته ای که در پیاده‌سازی این الگوریتم وجود دارد باید اندازه حتماً توانی از دو باشد. در صورتی که بخواهیم برای آرایه‌ای با اندازه دلخواه الگوریتم را پیاده‌سازی کنیم میتوانیم تعدادی صفر در انتهای آرایه اضافه کنیم به

صورتی که طور آرایه به اندازه کوچک ترین توان دو بزرگتر از اندازه اولیه شود. در این حالت المانهای بزرگتر از اندازه آرایه پس از محاسبه نتایج همگی ۰ خواهند بود و نتیجه‌ای در خروجی آرایه نخواهند داشت.