

به نام خدا

برنامه‌نویسی چندهسته‌ای

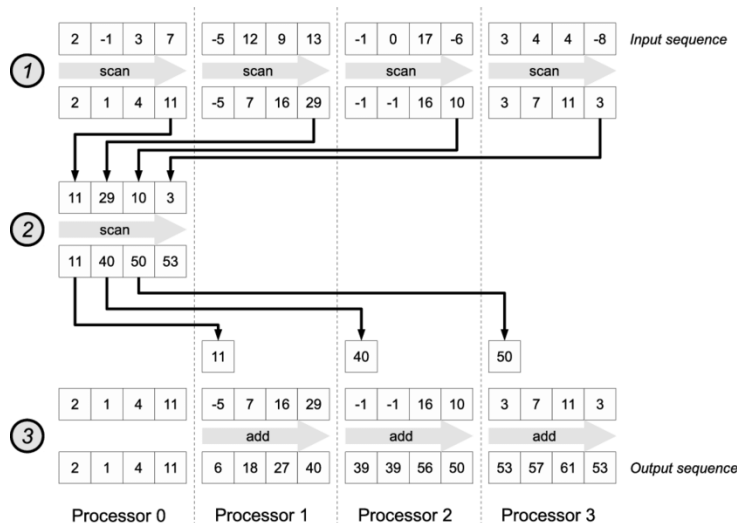
دستور کار آزمایشگاه ۶



پیش‌تر با موازی‌سازی Prefix sum در CPU آشنا شدیم. در این آزمایش قصد داریم تا الگوریتم‌های Prefix sum را بر روی GPU پیاده‌سازی کنیم. برای این کار ابتدا با الگوریتم‌های ساده شروع کرده سپس الگوریتم‌های پیچیده را پیاده‌سازی می‌کنیم. همانطور که می‌دانیم دو الگوریتم برای پیاده‌سازی روی CPU مورد بررسی قرار گرفت که قرار است روی GPU پیاده‌سازی شوند.

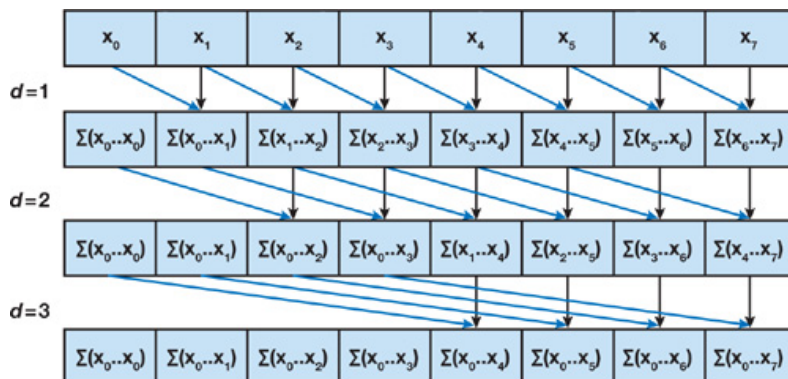
1- روش اول:

ابتدا آرایه را بین نخ‌ها به صورت static تقسیم کنید هر نخ عملیات Prefix sum را به صورت مستقل روی زیرآرایه خود انجام می‌دهد.



2- روش دوم:

الگوریتم Hillis and Steele که در سال 1986 برای محاسبه Prefix Sum معرفی شد. همانطور که قبلاً دیدیم این الگوریتم تسریع مناسبی روی CPU به ما نمی‌دهد.



گام اول:

ابتدا الگوریتم روش اول را روی GPU پیاده سازی کرده و نسبت به حالت اجرای موازی و سریال روی CPU آن را بررسی کنید.
کد اولیه پیاده سازی Prefix Sum در فایل `prefix_sum_serial.cpp` موجود هست.

1- آیا پیاده سازی این الگوریتم روی GPU بهتر است یا روی CPU با دلیل توضیح دهید؟

2- آیا افزایش اندازه کار هر warp تسریع را بهبود می بخشد؟

بار دیگر الگوریتم را با `shared memory` پیاده سازی کرده و تسریع و زمان اجرا را نسبت به حالت اولیه بررسی کنید.

گام دوم:

الگوریتم Hillis and Steele را روی GPU پیاده سازی کرده و تسریع و زمان اجرای آن را نسبت به روش اول موازی سازی روی CPU, GPU بررسی کنید.

3- تعداد و اندازه بلوک انتخابی خود را توجیه کنید.

a. آیا مقادیری که در نظر گرفته اید برای هر اندازه ورودی نیز بهتر از سایر مقادیر ممکن است؟

b. تعریف حافظه به شکل `pinned` یا `zero-copy` را برای رسیدن به سرعت بیشتر بررسی و تحلیل کنید.

4- زمان اجرای الگوریتم خود را برای آرایه هایی از نوع `int, float, double` مقایسه کنید و نتایج را توجیه کنید.

بار دیگر الگوریتم را با `shared memory` پیاده سازی کرده و تسریع و زمان اجرا را نسبت به حالت اولیه بررسی کنید.