

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

برنامه نویسی چندهسته‌ای

دستور کار آزمایشگاه ۴

هدف از این آزمایش، انجام عملیات Prefix sum بر روی یک آرایه است.

عملیات Prefix sum به صورت زیر تعریف می‌شود:

$$y[i] = \sum_{j=0}^i x[j]$$

به عبارت دیگر، هر درایه در آرایه خروجی، جمع همه درایه‌های قبل از خود در آرایه ورودی است.

این الگوریتم معمولاً به دو شیوه inclusive و exclusive پیاده می‌شود که تفاوت این دو را در ادامه می‌توانید ببینید:

Inclusive:

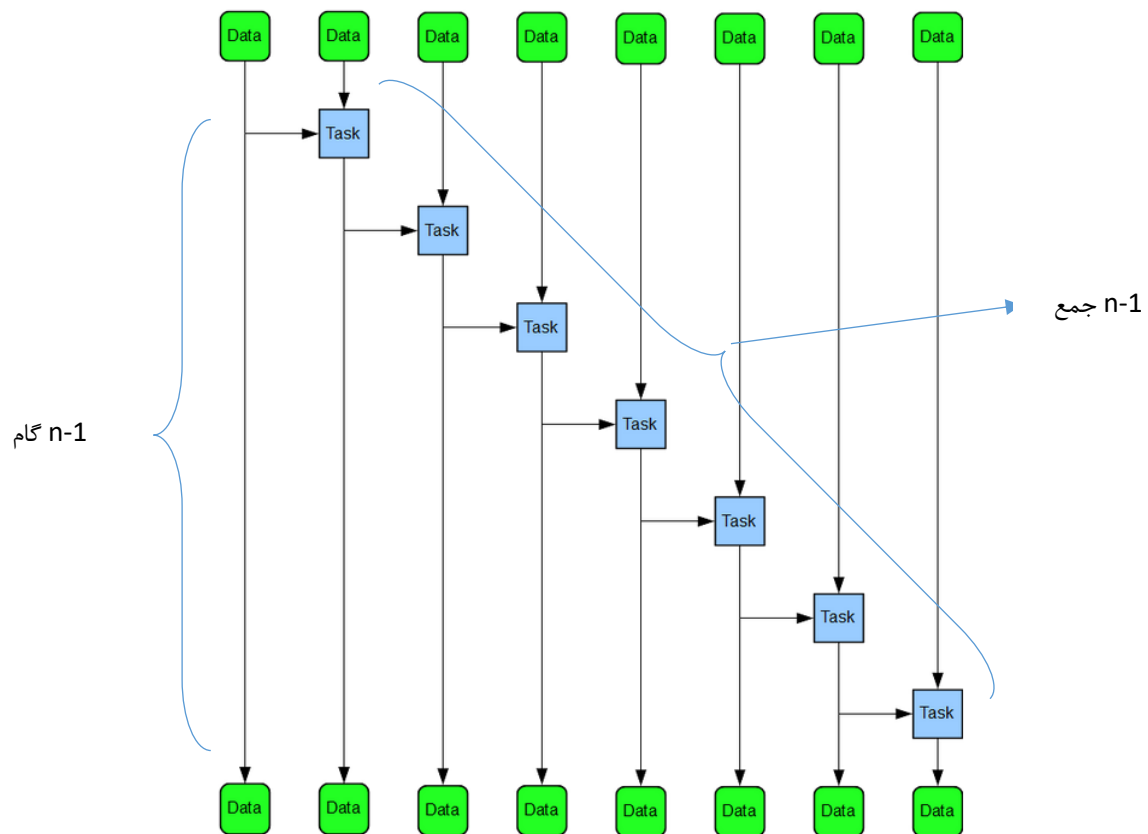
x	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
y	۱	۳	۶	۱۰	۱۵	۲۱	۲۸	۳۶	۴۵	۵۵

Exclusive:

x	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
y	۰	۱	۳	۶	۱۰	۱۵	۲۱	۲۸	۳۶	۴۵

کد سریال این الگوریتم با نام `lab_4_serial.c` در اختیار شما قرار دارد.

نحوه انجام این عملیات به صورت شماتیک در شکل زیر مشخص است:




در این آزمایش، دو روش موازی سازی الگوریتم Prefix sum معرفی می شود.

۱- روش اول:

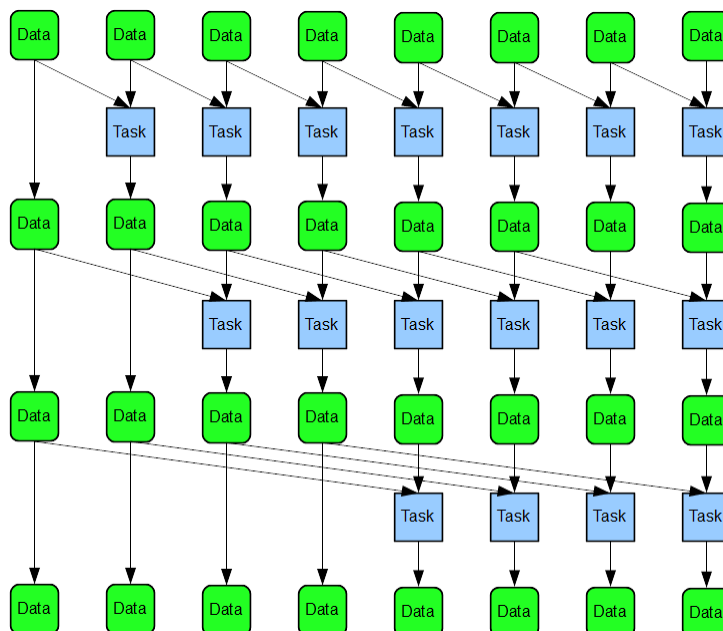
ابتدا آرایه x را بین نخ‌ها به صورت static تقسیم کنید (هر نخ $1/n$ آرایه را پردازش می‌کند). هر نخ عملیات prefix sum را به صورت مستقل بر روی زیرآرایه خود انجام می‌دهد. به عنوان مثال با دو نخ داریم:

x:	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
y:	۱	۳	۶	۱۰	۱۵	۶	۱۳	۲۱	۳۰	۴۰



- پس از انجام کار هر نخ باید مقدار خانه آخر هر زیر آرایه را با تمامی خانه‌های زیرآرایه‌های بعد از آن جمع کنیم. مثلاً مقدار خانه آخر زیرآرایه اول (۱۵) باید با همه خانه‌های زیرآرایه دوم جمع شود. چرا؟
- اگر در مثال بالا تعداد نخ‌ها چهار شود برای به دست آوردن مقدار نهایی یک‌راه حل این است که هر نخ مقدار آخرین خانه محاسبه شده توسط خود را با تمام خانه‌های پس از آن جمع کند. خروجی این راه حل را به ازای ۱ گیگ اندازه حافظه و ۴ نخ بررسی کنید. دلیل اشتباهات احتمالی چیست؟
- در روش قبل به ازای هر خانه از خروجی باید به تعداد بخش‌های قبل از آن روی خانه موردنظر عمل جمع صورت بگیرد. چطور می‌توان با انجام یک سری محاسبات میانی کم‌هزینه بر روی خانه‌های آخر زیر آرایه‌ها، به ازای هر زیرآرایه مقداری به دست آورد که اگر با خانه‌های آن زیرآرایه یکبار جمع شود خروجی نهایی مستقیماً به دست آید. کد موازی این روش را بنویسید و زمان اجرای آن را به ازای ۱ گیگ اندازه حافظه با حالت سریال مقایسه کنید.
- (امتیازی): اگر ایده‌ای دارید که سریع‌تر به جواب می‌رسد آن را پیاده‌سازی کنید.

یک الگوریتم برای محاسبه prefix scan الگوریتمی است به نام Hillis and Steele که در سال ۱۹۸۶ معرفی شده است. شکل زیر الگوی محاسبات آن را نشان می‌دهد:



در این شکل، آرایه ورودی دارای ۸ المان است و آرایه خروجی در پایین محاسبه شده است. هر مربع task یک جمع است.

- کد موازی این الگوریتم را بنویسید و زمان اجرای آن را به ازای ۱ گیگ اندازه حافظه با حالت سریال مقایسه کنید.
- الگوریتم دوم با اینکه کار موازی بیشتری تولید می‌کند ولی کندتر از الگوریتم اول و حتی الگوریتم سریال است. چرا؟
- توضیح دهید الگوریتم دوم در چه حالتی می‌تواند نسبت به الگوریتم اول مزیت داشته باشد (راهنمایی: این الگوریتم در GPU بسیار پر کاربرد است).

عملیات prefix sum کاربرد فراوانی در حوزه‌های مختلف دارد. چند نمونه از آن‌ها را ذکر کنید.

توجه: در گزارش از هر مرحله کد خود اسکرین‌شات قرار دهید یا فایل کد هر مرحله را به صورت جداگانه آپلود کنید.