2018

MC-HW4

ساره سلطانی نژاد ۹۳۳۱۰۳۹

سوال ۱

سرعت اتصال حافظه‌های جانبی در دو معماری چه تفاوتی دارد؟

با توجه به شکل در معماری شکل ۱ سرعت اتصال حافظه‌های جانبی برابر 300 MB/S می‌باشد

در شکل دوم سرعت اتصال برابر 600 MB/S است.

آیا سرعت اتصال حافظه‌های جانبی اهمیت دارد؟ چرا؟

بله. اگه بخواهیم سیستم ما سریع باشد لازم است سرعت خواندن داده از حافظه و نوشتن نتیجه در آن به همان نسبت بالا باشد، هم چنین نیاز به گذرگاهی با سرعت بالا داریم. می‌دانیم سرعت سیستم تابعی از سرعت بخش-های مختلف آن است لذا اگر بخشی کند باشد، موجب میشد میزان سرعت کل سیستم پایین بیاید لذا باید برای گرفتن تسریع کلی باید سرعت همه بخش‌ها را بهبود ببخشیم

برای اتصال GPU به سیستم از چه درگاهی استفاده میشود؟ ویژگیهای این درگاه چیست؟

برای اتصال تمام دیستگاه‌های GPU به سیستم از PCI-E bus استفاده می‌شود. در این مثال bus از نوع PCI-E 2.0 می‌باشد که سریع‌ترین باس موجود است و نرخ انتقال آن ۵ گیگابایت بر ثانیه می‌باشد. امروزه bus PCI-E 3.0 معرفی شده است که مقدار پهنای باند در دسترس را به طور قابل ملاحظه‌ای افزایش داده است. PCI-E 2.0 بر خلاف PCI بر مبنای پهنای باند تضمین شده‌ای کار می‌کند. در سیستم‌های PCI قبلی، هر جز میتوانست از کل پهنای باند bus استفاده بکند اما تنها یک دستگاه در هر زمان قادر به انجام این کار بود که موجب می‌شد به هر چه کارت بیشتری شما به سیستم اضافه کنید، پهنای باند کمتری برایهر کارت در دسترس بود که باس PCI-E 2.0 با معرفی lane هایی این مشکل را حل نمود. این‌ها لینک‌های سریال سریعی هستند که می‌توانند با یکدیگر ترکیب شده و به فرم X1, X2, X4, X8, X16 می‌باشند. امروزه بسیاری از GPU ها از باس PCI-E 2.0, X16 استفاده می‌کنند. این باس ارتباطش از نوع full- duplex است و نرخ انتقال آن ۵ گیگابایت بر ثانیه می‌باشد که به این معنا است که میتوانیم سرعت ارسال و دریافت یکسانی در یک زمان واحد داشته باشیم. پس میتوان با نرخ ۵ گیگابایت بر ثانیه داده به کارت ارسال کرد و با همین نرخ هم داده از آن دریافت کرد. ولی این به معنا نیست که اگر ما داده‌ای از کارت دریافت نکردیم میتوانیم با نرخ ۱۰ گیگا بایت بر ثانیه اطلاعات ارسال کنیم.

با توجه به شکل‌های نشان داده شده چند GPU و با چه پهنای باندی میتوان در هر معماری به سیستم متصل کرد؟

معماری شکل ۱:

در این معماری با x58 و سوکت پردازنده 1366 در مجموع ۳۶ خط PCI-E در دسترس است که به این معناست که با x16 دو کارت (GPU) و با x8 چهار کارت (GPU) را پشتیبانی می‌کند.

طراحی x58 در تراشه lesser P55 با ۱۶ خط در دسترس است که به این معناست یک کارت GPU در x16 (که کل پهنای باند را در اختیار می‌گیرد) و دو کارت در x8 میتوان به سیستم متصل نمود.

معماری شکل ۲:

با طراحی تراشه 17/X58 اینتل به سمت طراحی Sandybridge رفت که یکی از پیشرفت‌های این طراحی حمایت از استاندارد SATA-3 می‌باشد که از نرخ انتقال ۶۰۰ مگابایت بر ثانیه پشتیبانی می‌کند. طراحی Sandybridge تنها از ۱۶ خط PCI-E پشتیبانی می‌کند(که در این صورت میتوان با X16 تنها یک کارت (GPU) را به سیستم متصل نمود) و پهنای باند را به 16 GB/S در تئوری و 10 GB/s در عمل محدود می‌کند. در Sandybridge-E تعدا خطوط PCI-E برابر ۴۰ می‌باشد

سوال ۲

a : GPU در واقع ارایه ای از SMها می‌باشد که هر کدام N هسته دارد (برای مثال ۸ تا در G80, GT200 و ۳۲-۴۸ تا در Fermi و بیشتر از ۸ تا در kepler) که این خود علت اصلی مقیاس پذیری پردازنده‌ها می‌باشد. یک دستگاه GPU از یک یا تعداد بیشتری SM تشکیل شده است. اضافه کردن تعداد بیشتری SM به دستگاه موجب قادر کردن GPU به پردازش تعداد بیشتری Task در یک زمان ثابت یا اجرای سریعتر یک Task، در صورت وجود قابلیت موازی سازی در تسک، می‌کند.

هر SM از تعدادی اجزای اصلی تشکیل شده است. هر SM از تعدادی SPs تشکیل شده است که تعداد آن‌ها در معماری‌های مختلف متفاوت است. مثلا در Fermi ۳۲-۴۸ تا SP وجود دارد و در kepler این تعداد به ۱۹۲ می-رسد. تعداد SMs, SPs در نسل های بعدی سخت افزار در حال افزایش خواهد بود.

هر SM به بخشی به نام رجیستر فایل دسترسی دارد که مشابه بخشی از حافظه است و با سرعتی برابر سرعت واحد SP کار می‌کند که این موجب می‌شود زمان انتظار در این حافظه صفر باشد. اندازه این حافظه در نسل‌های مختلف متفاوت است. رجستر فایل برای ذخیره رجیسترهای مورد استفاده تردهای در حال اجرا استفاده می‌شود.

همچنین یک بلاک حافظه مشترک هم وجود دارد که تنها توسط SMها در دسترس است که می‌تواند به عنوان یک حافظه نهان مدیریت شده استفاده بشود. بر خلاف کش CPU هیچ سخت افزاری نمی‌تواند از این حافظه داده خارج بکند و حافظه تماما تحت کنترل برنامه‌نویس است.

هر SM باس‌های جداگانه‌ای برای حافظه‌های texture، constant و global دارد. حافظه texture یک شمای خاص از حافظه global است که برای حالتی که بین داده‌ها تعامل وجود دارد، مناسب است. برای مثال برای جدول های lookup دو بعدی یا سه بعدی، این حافظه ویژگی خاصی مبتنی بر سخت افزار برای تعامل دارد.

حافظه constant ، برای داده‌های فقط خواندنی استفاده میشود و روی سخت افزار cache می‌شود. همانند حافظه texture، حافظه constant نیز نمای سادهای از حافظه global است.

b : حافظه سراسری توسط GDDR روی کارت گرافیک تغذیه می‌شود. این یک ورژن با کارآیی و عملکرد بهتر نسبت به DDR است. پهنای گذرگاه حافظه می‌تواند تا ۵۱۲ بیت باشد که پهنای باندی ۵ تا ۱۰ برابر پهنای باند cpu است و GPU به این پهنای باند بالا نیاز دارد و به همین دلیل از GDDR استفاده می‌کند

c : هر SM از تعدادی اجزای اصلی تشکیل شده است. هر SM از تعدادی SPs تشکیل شده است که تعداد آن‌ها در معماری‌های مختلف متفاوت است. مثلا درFermi ۳۲-۴۸ تا SP وجود دارد و در kepler این تعداد به ۱۹۲ می‌رسد. تعداد SMs, SPs در نسل های بعدی سخت افزار در حال افزایش خواهد بود.
هر SM به بخشی به نام رجیستر فایل دسترسی دارد که مشابه بخشی از حافظه است و با سرعتی برابر سرعت واحد SP کار می‌کند که این موجب می‌شود زمان انتظار در این حافظه صفر باشد. اندازه این حافظه در نسل‌های مختلف متفاوت است. رجستر فایل برای ذخیره رجیسترهای مورد استفاده تردهای در حال اجرا استفاده می-شود. همچنین یک بلاک حافظه مشترک هم وجود دارد که تنها توسط SMها در دسترس است که می‌تواند به عنوان یک حافظه نهان مدیریت شده استفاده بشود. بر خلاف کش CPU هیچ سخت افزاری نمی‌تواند از این حافظه داده خارج بکند و حافظه تماما تحت کنترل برنامه‌نویس است.
هر SM باس‌های جداگانه‌ای برای حافظه‌های texture، constant و global دارد.
هر SM دارای دو یا بیشتر واحد خاص منظوره SPU است که هر کدام، دستورالعمل‌های سخت افزاری خاصی را اجرا می‌کنند.

d : هر SM باس‌های جداگانه‌ای برای حافظه‌های texture، constant و global دارد و از لحاظ فیزیکی جدا هستند. حافظه texture یک شمای خاص از حافظه global است که برای حالتی که بین داده‌ها تعامل وجود دارد، مناسب است. برای مثال برای جدول های lookup دو بعدی یا سه بعدی، این حافظه ویژگی خاصی

مبتنی بر سخت افزار برای تعامل دارد. حافظه constant ، برای داده‌های فقط خواندنی استفاده می‌شود و روی سخت افزار cache می‌شود. همانند حافظه texture، حافظه constant نیز نمای سادهای از حافظه global است.

سوال ۳:

Compute Capability : مجموعه ویژگی‌هایی سخت افزاری و نرم افزاری مربوط به محاسبات که سخت افزار CUDA پشتیبانی می‌کند را شرح می‌دهد. در ورژن‌های مختلف Computing Capability : طراحی سخت افزار، تعداد هسته‌ها، اندازه cache، دستورالعمل‌های ریاضی پشتیبانی شده، متفاوت است. Computing Capability با یک شماره نسخه مشخص می‌شود که SM Version نامیده می‌شود. وقتی با cuda برنامه می‌نویسیم، این خیلی مهم است که به تفاوت‌های بین سخت افزارهای مختلف توجه کنیم. برای مثال Computing Capability 1.x دارای ۱۶ کیلوبیت حافظه محلی برای هر ترد است در حالیکه Computing Capability 2.x, Computing Capability 3.x دارای ۵۱۲ کیلوبیت حافظه محلی برای هر ترد است.

https://cvw.cac.cornell.edu/gpu/computecap

سوال ۴:

Occupancy به صورت نسبت تعداد wrapهای فعال در SM به حداکثر تعداد wrapهایی که در آن SM پشتیبانی می‌شود، تعریف می‌شود. یک wrap را از زمانی که تردها شروع به اجرا می‌کنند تا زمانی که تمام تردها از کرنل خارج شوند، را فعال در نظر می‌گیرند.

Occupancy = # of active warps / Maximum number of resident warps per SM

Occupancy در واقع معیاری مرتبط با تعداد wrapهای فعال می‌باشد که با توجه به آن می‌فهمیم سخت‌افزار چقد مشغول نگه داشته شده است. و درصد توانایی سخت افزار برای پردازش wrapهای فعال را نشان می‌دهد.

Occupancy با توجه به مولفه‌های زیر محاسبه می‌شود:

- ✓ Blocks per SM
- ✓ Warps per SM
- ✓ Registers per SM
- ✓ Shared Memory per SM

https://docs.nvidia.com/gameworks/content/developertools/desktop/analysis/report/cudaexperiments/kernellevel/achievedoccupancy.htm

سوال ۵:

هر نخ ابتدا باید با استفاده از اندیس‌های خود مشخص کند که باید بر روی چه قسمتی از داده کار کند. لذا ابتدا باید ببیند در کدام بلاک است و سایز هر بلاک چقدر است و به تعداد خانه‌های بلاک‌های قبل از خود جلو بیاید(blockIdx.x*blockDim.x) و حال با توجه به اندیس خود(threadIdx.x) می‌تواند بفهمد در بلاک خود در کدام خانه است. لذا گزینه c درست می‌باشد

```
I  = blockIdx.x*blockDim.x + threadIdx.x;
```

سوال ۶ :

گزینه c درست می‌باشد

با توجه به اینکه هر نخ یک خروجی را تولید می‌کند نیاز به ۸۰۰۰ نخ داریم. برای اینکه تعداد بلوک‌ها حداقل باشد نیاز داریم حداکثر تعداد نخ‌ها در هر بلو ک داشته باشیم. با توجه به سایز بلوک که ۱۰۲۴ است تعداد نخ‌ها هم باید بر ۱۰۲۴ بخش پذیر باشد لذا کمترین مقدار برابر ۸۱۹۲ است که بر ۱۰۲۴ بخش پذیر است و ۸۰۰۰ نخ در اختیار ما قرار می‌دهد.

1024 * 8 = 8192 > 8000