به نام خدا

تمرین چهارم درس برنامه نویسی چند هسته ای

امیرمحمد پیرحسین لو

۹۵۳۱۰۱۴

۱- پاسخ سوال اول:

a. سرعت اتصال حافظه ها در معماری Nehalem بیشتر از معماری Sandybridge است.(در معماری Nehalem از QPI یا همان Quick Path Interconnect استفاده شده است.)

b. بله اهمیت دارد. به دلیل اینکه حافظه ها مستقیم به GPU متصل نیستند، برای انتقال داده بین memory و GPU باید از درگاه اتصال دهنده memory به CPU استفاده کرد که هر چه قدر سرعتش بیشتر باشد سرعت انتقال داده بیشتر می شود.

c. درگاه PCI_E. درگاهی پر سرعت است که device های وصل شده به آن به صورت هم زمان می توانند داده دریافت کنند یا انتقال دهند. این درگاه از واحد هایی به نام lane تشکیل شده است و از ترکیب آن ها می توان لینک های X1, X2, X4, X8 و X16 را به وجود دارد. سرعت دانلود و اپلود در آن برابر است و می توان همزمان هر دو عمل را انجام داد.

d. در معماری Nehalem، PCIE موجود ۳۶ lane دارد. اگر GPU ها دارای لینک X16 باشند، حداکثر ۲ GPU می توان به سیستم متصل کرد.

در معماری Sandybridge، PCIE موجود ۱۶ lane دارد. اگر GPU ها داری لینک X16 باشند، حداکثر ۱ GPU می توان به سیستم متصل کرد.

۲- پاسخ سوال دوم:

a. هر GPU تعدادی SM دارد و هر SM با توجه به معماری آن دارای تعدای هسته است که دستورالعمل ها را اجرا می کنند (مثلا در معماری G80 تعداد هسته ها 8 است.). همه ی هسته ها در هر لحظه عملیاتی یکسان اجرا می کنند. هر SM دارای تعدادی SP است (مثلا معماری Fermi بین 32 تا 48 تا SP دارد.). وظیفه SP انقال داده ها و دستورالعمل به core ها برای اجرا است. دستورات برای اجرا نیاز به داده دارند که این داده ها در global memory ذخیره می شود. هر GPU یک global memory دارد و همه ی SM ها می توانند به آن دسترسی پیدا کنند، در آن write کنند یا از آن بخوانند. از طرفی دیگر هر SM یک shared memory در اختیار دارد که تنها توسط block های موجود در آن قابل دسترسی است و فضای آن بین block ها تقسیم می شود. شباهت زیادی بین shared memory و cache وجود دارد ولی یک تفاوت عمده نیز وجود دارد و آن نیز این است که کنترل shared memory کامل در اختیار برنامه نویس است در صورتی که کنترل cache در اختیار برنامه نویس نیست. texture memory یک دید نسبت به global memory فراهم می آورد و برای 2D lookup table و 3D lookup table از آن استفاده می شود. constant memory هم یک نوع cache است که فقط قابل خواندن است و مانند texture memory یک دید نسبت به global memory در اختیار ما می دهد. (فضای آن جدا از global memory نیست.)

b. GDDR مخفف Graphic Double Data Rate است. یک ورژن جدیدتر از DDR است که کارایی بسیار بالایی دارد. پهنای باند آن بین ۵ تا ۱۰ برابر بیشتر از پهنای باند بین memory و CPU است. به دلیل وجود تعدای زیادی core برای تغذیه آن ها نیاز به پهنای باند بالا است. معماری GPU از نوع SIMD است و این یعنی به ازای هر دستورالعمل

تعداد زیادی داده باید واکشی شود و برای اینکه همچنان کارایی بالا باشد باید پهنای باند بالایی برای خواندن داده از memory در اختیار داشته باشیم. به همین دلیل از GDDR جای DDR استفاده می شود.

c. شرح اجزا:

i. global memory: حافظه مشترک بین SM ها

ii. L1 and L2 cache: همانند cache سطح 2 و 1 سی پی یو عمل می کند.

iii. Texture Cache و Constant Cache: در قسمت الف در مورد آن ها توضیح داده شد.

iv. SM: واحدی که از نظر فیزیکی شامل تعداد زیادی هسته است که دستورالعمل ها را اجرا می کنند. از نظر نرم افزاری می توان تعدادی بلاک به آن اختصاص داد که هر بلاک شامل تعدای نخ است. نخ ها به دسته هایی تقسیم می شوند ک به آن ها warp گفته می شود. هر دسته نخ در لحظه دستورالعمل واحدی را روی تعدادی داده اجرا می کند.

v. SP: واحدی که شامل تعداد زیادی هسته است و نخ ها روی این هسته ها دستورالعمل ها را اجرا می کنند.

vi. register file: SM دارای تعداد زیادی register به شدت پر سرعت است که دسترسی به آن ها یک سیکل ساعت طول می کشد. این رجیسترها بین نخ ها تقسیم می شود و نخ ها به کمک آن ها کارهای خود را انجام می دهند.

vii. SPU یا special purpose unit: واحدی که عملیات های خاص مانند محاسبه sin و cos را انجام می دهد.

d. حافظه Constant تنها خواندنی است. حافظه های Constant و Texture حجم خیلی کمتری نسبت به global memory دارند و عملکرد آن ها به عملکرد cache شبیه تر است. از نظر فیزیکی جدا نیستند. حافظه های Constant و Texture یک دید دیگر نسبت به global memory هستند.

۳- پاسخ سوال سوم: compute capability ویژگی هایی را که توسط یک سخت افزار CUDA پشتیبانی می شود بیان می کند. برای مثال، GeForce GTX 680 شامل ۱۵۳۶ هسته و compute capability برابر 3.0 است.

۴- occupancy برابر میاتگین درصد تعداد warp های اکتیو در زمان اجرای kernel است. active warp به warp ای گفته می شود که نخ های آن در حال اجرا هستند و در مودهای دیگر مانند ready و انتقال داده با IO قرار ندارد. برای محاسبه آن باید بیشینه تعداد warp های قابل فعال سازی در SM را بدانیم که به تعداد warp scheduler ها و همچین نوع برنامه (کاربرد) وابسته است.

۵- c. blockIdx.x * blockDim.x + threadIdx.x

۶- c. 8192