



请输入关键词

首页

编程语言

Web前端

系统架构

数据库

移动开发

操作系统

开源软件

互联网

行业应用

研发管理

IT生活

论坛

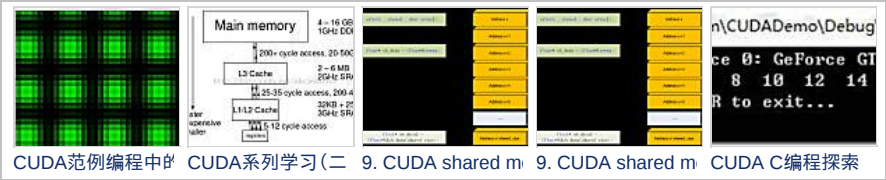
当前位置: 首页 > 资讯 > info5 > 正文

CUDA编程——Memory Coalescing

发表于: 2016-01-18 作者: junparadox 来源: 转载 浏览: 79

CUDA 编程 GPU

分享到: 新浪微博 微信 腾讯微博 人人网 有道云笔记 QQ空间



摘要: CUDA编程——MemoryCoalescing1GPU总线寻址介绍假定X是一个指向整数(32位整数)数组的指针, 数组的首地址为0x00001232。一个线程要访问元素X[0],.inttmp=X[0];假定memory总线宽度为256位, 因为基于字节地址的总线要访问memory, 必须和总线宽度对齐, 也就是说按必须32字节对齐来访问memory, 比如访问0x00000000,0x00000020,

CUDA编程——Memory Coalescing

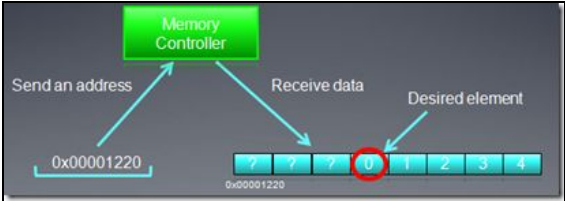
1 GPU总线寻址介绍

假定X是一个指向整数(32位整数)数组的指针, 数组的首地址为0x00001232。一个线程要访问元素X[0],



int tmp = X[0];

假定memory总线宽度为256位, 因为基于字节地址的总线要访问memeory, 必须和总线宽度对齐, 也就是说按必须32字节对齐来访问memory, 比如访问0x00000000,0x00000020,0x00000040,...等, 所以我们要得到地址0x00001232中的数据, 比如访问地址0x00001220,这时, 它会同时得到0x00001220到 0x0000123F 的所有数据。只是对我们来说, 只有一个32位整数有用, 所以有用的数据是4个字节, 其它28的字节的数据都被浪费了, 白白消耗了带宽。



2 合并内存访问

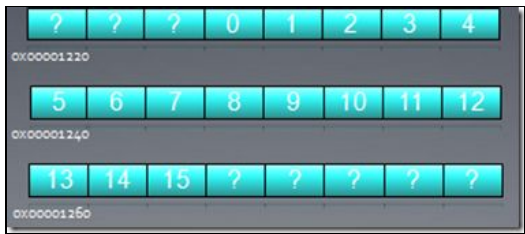
为了利用总线带宽, GPU通常把多个线程的访存合并到较少的内存请求中去。假定多个线程访问32个字节以内的地址, 它们的访问可以通过一个memory request完成, 这样可以大大提高带宽利用率, 在专业术语描述中这样的合并访问称作coalescing。

推荐文章

- 1 cuda memory
- 2 CUDA ---- Shared Memory
- 3 CUDA ---- Memory Model
- 4 CUDA ---- Memory Access
- 5 CUDA Texture Memory
- 6 CUDA Shared Memory : transpose
- 7 CUDA Texture Memory
- 8 CUDA Shared Memory : transpose
- 9 CUDA shared memory
- 10 并行程序设计---cuda memory
- 11 cuda编程: 关于共享内存 (shared
- 12 CUDA Texture Part.2 Linear Mem
- 13 CUDA 全局global memory变量
- 14 [CUDA]CUDA C并行编程
- 15 6.1 CUDA: pinned memory固定存储
- 16 cuda的Pinned Memory (分页锁定
- 17 CUDA C编程入门
- 18 CUDA编程入门
- 19 CUDA编程学习(一)
- 20 CUDA编程学习(二)

编辑推荐

- 1 CUDA范例编程中的shaed memory b
glut32.lib放到C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v4.1\lib\Win32 (根据安装的目的)
- 2 CUDA系列学习(二)CUDA memory
本文来介绍CUDA的memory和变量存放, 分为以下章节: (一)、CPU Memory 结构 (二)、GPU Memory结构
- 3 9. CUDA shared memory使用-----
9. CUDA shared memory使用-----GPU的革命 序言: 明年就毕业了, 下半年就要为以后的生活做打算。这
- 4 9. CUDA shared memory使用-----
9. CUDA shared memory使用-----GPU的革命 序言: 明年就毕业了, 下半年就要为以后的生活做打算。这
- 5 CUDA C编程探索
摘要: 本文论述了使用CUDA C编写Windows Console Application、动态链接库(DLL)、在 .NET 中使用CUD
- 6 CUDA编程札记
const int N = 33 * 1024; const int threadsPerBlock = 256; const int blocksPerGrid = imin(32,
- 7 cuda编程知识普及
本帖经过多方整理, 大多来自各路书籍《GPGPU编程技术》《cuda高性能》1 grid 和 block都可以用三元
- 8 CUDA编程模型
CUDA编程模型 CUDA将CPU作为主机(Host), GPU作为设备(Device)。一个系统中可以有一个主机和多个
- 9 CUDA编程札记
const int N = 33 * 1024; const int threadsPerBlock = 256; const int blocksPerGrid = imin(32,



例如上面16个线程访问地址0x00001232 到 0x00001272, 我们只需要3次memory request。

3 Bank Conflicts

对Nvidia GPUs来说, local memory是由banks组成的, 每个bank是32bit, 可视化图如下。bank是实际存储单元。每个bank在一次访存中, 可以被取址一次。并行访问相同的bank, 将导致访存串行 (bank conflicts)。

Bank		1		2		3		...
Address		0 1 2 3		4 5 6 7		8 9 10 11		...
Address		64 65 66 67		68 69 70 71		72 73 74 75		...

CUDA编程中,一个half-warp (16个threads) 访问连续的32bit地址,不会有bank conflicts。一个例外情况是broadcast, 如果所有thread访问同一个地址, 内存只会被读一次, 并broadcast到所有threads。

CUDA编程——Memory Coalescing

0票

0票

0票

0票

0票

0票

0票

0票

开心 板砖 感动 有用 疑问 难过 无聊 震惊

0

0

顶

踩

