

公告

昵称：mydddfly  
园龄：2年1个月  
粉丝：11  
关注：12  
+加关注

<	2018年1月						>
日	一	二	三	四	五	六	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	

搜索

找找看

谷歌搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

GPU：并行计算利器

http://blog.jobbole.com/87849/



- [首页](#)
- [最新文章](#)
- [IT 职场](#)
- [前端](#)
- [后端](#)
- [移动端](#)
- [数据库](#)
- [运维](#)
- [其他技术](#)

- 导航条 -

▼

## 我的标签

[android\(71\)](#)

[图像处理\(65\)](#)

[php\(39\)](#)

[ai\(23\)](#)

[连接池\(22\)](#)

[流媒体\(20\)](#)

[地址抓取\(17\)](#)

[mysql\(17\)](#)

[opencv\(15\)](#)

[opengl\(13\)](#)

[更多](#)

## 随笔档案

[2018年1月 \(1\)](#)

[2017年11月 \(37\)](#)

[2017年10月 \(7\)](#)

[2017年9月 \(3\)](#)

[2017年8月 \(12\)](#)

[2017年7月 \(33\)](#)

[2017年6月 \(38\)](#)

[2017年5月 \(14\)](#)

[2017年4月 \(13\)](#)

[2017年3月 \(14\)](#)

[2017年2月 \(6\)](#)

[2017年1月 \(6\)](#)

[2016年11月 \(14\)](#)

[2016年10月 \(5\)](#)

[2016年9月 \(11\)](#)

[2016年8月 \(14\)](#)

[2016年7月 \(33\)](#)

[伯乐在线](#) > [首页](#) > [所有文章](#) > [IT技术](#) > GPU：并行计算利器

# GPU：并行计算利器

2015/06/28 · [IT技术](#) · [GPU](#), [并行计算](#)

分享到：[31](#)

原文出处：[占利军 的博客](#)

## 1 GPU是什么

如图1所示，这台PC机与普通PC机不同的是这里插了7张显卡，左下角是显卡，在中间的就是GPU芯片。显卡的处理器称为图形处理器（GPU），它是显卡的“心脏”，与CPU类似，只不过GPU是专为执行复杂的数学和几何计算而设计的。

GPU计算能力非常强悍，举个例子：现在主流的i7处理器的浮点计算能力是主流的英伟达GPU处理器浮点计算能力的1/12。

2016年6月 (5)  
2016年5月 (23)  
2016年4月 (25)  
2016年3月 (50)  
2016年2月 (27)  
2016年1月 (6)

## 最新评论

1. Re:php 中处理 websocket  
@爱吃鱼的大大脸猫收费的吧...  
--\_York
2. Re:php 中处理 websocket  
web实时推送技术使用越来越广泛，  
但是自己开发又太麻烦了，我觉得  
没有那个必要，GoEasy就挺不错  
的，服务器稳定，代码简洁易懂；  
官网：  
--爱吃鱼的大大脸猫
3. Re:使用PHP连接、操纵  
Memcached的原理和教程  
学习了  
--源源猿
4. Re:关于MySQL中使用LOAD  
DATA INFILE导入csv文件时的日  
期格式问题  
老哥 我有两个时间字段的时候该怎么  
用呢？我刚刚用了报错了，能不能  
帮我看一下啊，下面是源码load  
data infile  
'C:/ProgramData/MySQL/MyS  
QL Server 5.7.....

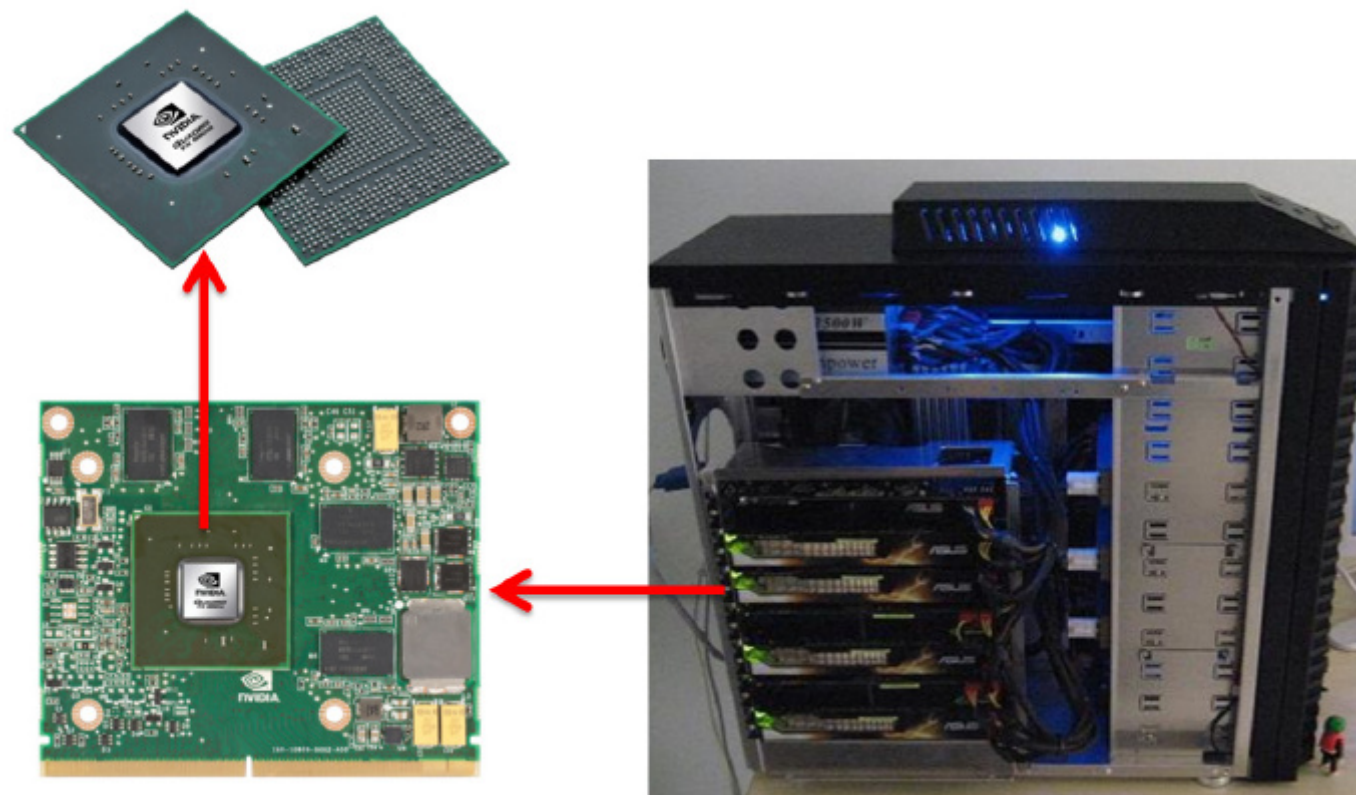


图1 显卡与GPU

## 2 为什么GPU计算能力如此强悍？

图2对CPU与GPU中的逻辑架构进行了对比。其中Control是控制器、ALU算术逻辑单元、Cache是cpu内部缓存、DRAM就是内存。可以看到GPU设计者将更多的晶体管用作执行单元，而不是像CPU那样用作复杂的控制单元和缓存。从实际来看，CPU芯片空间的5%是ALU，而GPU空间的40%是ALU。这也是导致GPU计算能力超强的原因。

## 阅读排行榜

1. MySql 存储过程实例(附完整注释)(19798)
2. Apache Rewrite url重定向功能的简单配置(8989)
3. PHP json\_encode() 函数介绍 (7608)
4. html5 video.js 使用及兼容所有浏览器(6194)
5. php连接Access数据库的三种方法(6172)

## 评论排行榜

1. php 中处理 websocket(2)
2. 关于MySQL中使用LOAD DATA INFILE导入csv文件时的日期格式问题(1)
3. 使用PHP连接、操纵Memcached的原理和教程(1)

## 推荐排行榜

1. Nginx负载均衡配置实例详解(2)
2. 关于MySQL中使用LOAD DATA INFILE导入csv文件时的日期格式问题(1)
3. WebSocket 实战(1)
4. 使用PHP连接、操纵Memcached的原理和教程(1)

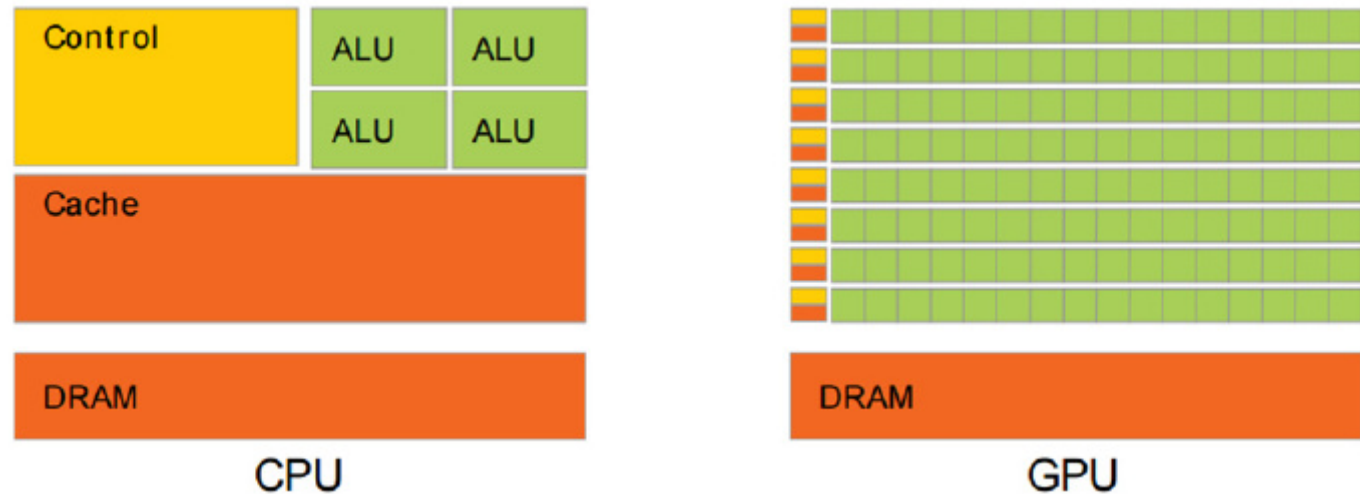


图2 cpu和gpu硬件逻辑结构对比

那有人讲了，为什么cpu不像gpu那样设计呢，这样计算能力也强悍了！

为什么？CPU要做得很通用。CPU需要同时很好的支持并行和串行操作，需要很强的通用性来处理各种不同的数据类型，同时又要支持复杂通用的逻辑判断，这样会引入大量的分支跳转和中断的处理。这些都使得CPU的内部结构异常复杂，计算单元的比重被降低了。而GPU面对的则是类型高度统一的、相互无依赖的大规模数据和不需要被打断的纯净的计算环境。因此GPU的芯片比CPU芯片简单很多。

举个例子，假设有一堆相同的加减乘除计算任务需要处理，那把这个任务交给一堆（几十个）小学生就可以了，这里小学生类似于GPU的计算单元，而对一些复杂的逻辑推理等问题，比如公式推导、科技文章写作等高度逻辑化的任务，交给小学生显然不合适，这时大学教授更适合，这里的大学教授就是CPU的计算单元了，大学教授当然能处理加减乘除的问题，单个教授计算加减乘除比单个小学生计算速度更快，但是成本显然高很多。

## 3 GPU编程库

GPU计算能力这么强，被广泛使用！比如挖矿（比特币）、图形图像处理、数值模拟、机器学习算法训练等等，那我们怎么发挥GPU超强的计算能力呢？一编程！

怎么进行GPU编程呢？现在GPU形形色色，比如Nvidia、AMD、Intel都推出了自己的GPU，其中最为流行的就是Nvidia的GPU，其还推出了CUDA并行编程库。然而每个GPU生产公司都推出自己的编程库显然让学习成本上升很多，因此苹果公司就推出了标准

5. 缓存(之一) 使用Apache Httpd实现http缓存(1)

OpenCL，说各个生产商都支持我的标准，只要有一套OpenCL的编程库就能对各类型的GPU芯片适用。当然了，OpenCL做到通用不是没有代价的，会带来一定程度的性能损失，在Nvidia的GPU上，CUDA性能明显比OpenCL高出一大截。目前CUDA和OpenCL是最主流的两个GPU编程库。

从编程语言角度看，CUDA和OpenCL都是原生支持C/C++的，其它语言想要访问还有些麻烦，比如Java，需要通过JNI来访问CUDA或者OpenCL。基于JNI，现今有各种Java版本的GPU编程库，比如JCUDA等。另一种思路就是语言还是由java来编写，通过一种工具将java转换成C。

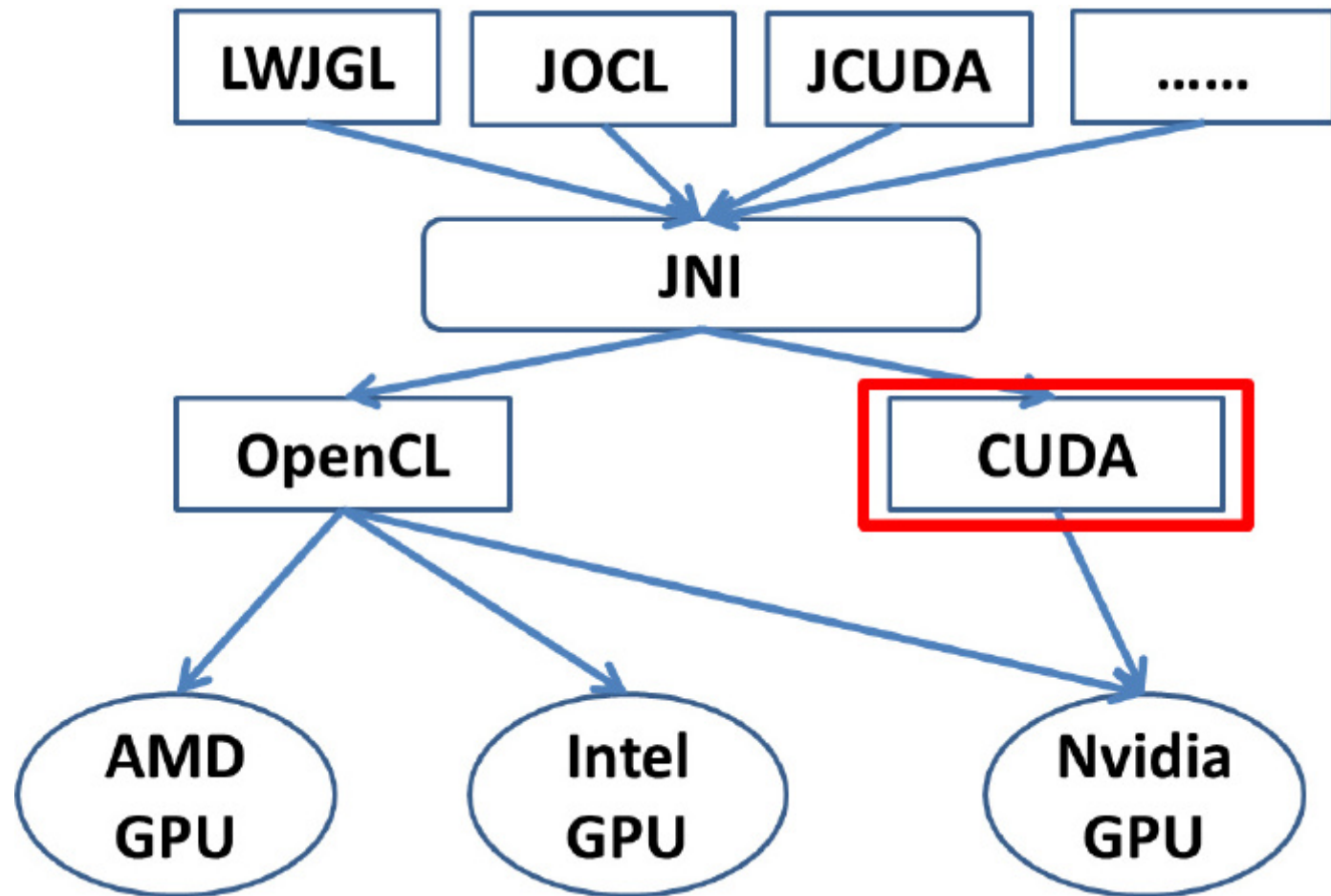


图3 GPU编程库

LWJGL (<http://www.lwjgl.org/>)

JOCL (<http://www.jocl.org/>)

JCUDA (<http://www.jcuda.de/>)

Aparapi (<http://code.google.com/p/aparapi/>)

JavaCL (<http://code.google.com/p/javaccl/>)

## 4 CUDA程序流程

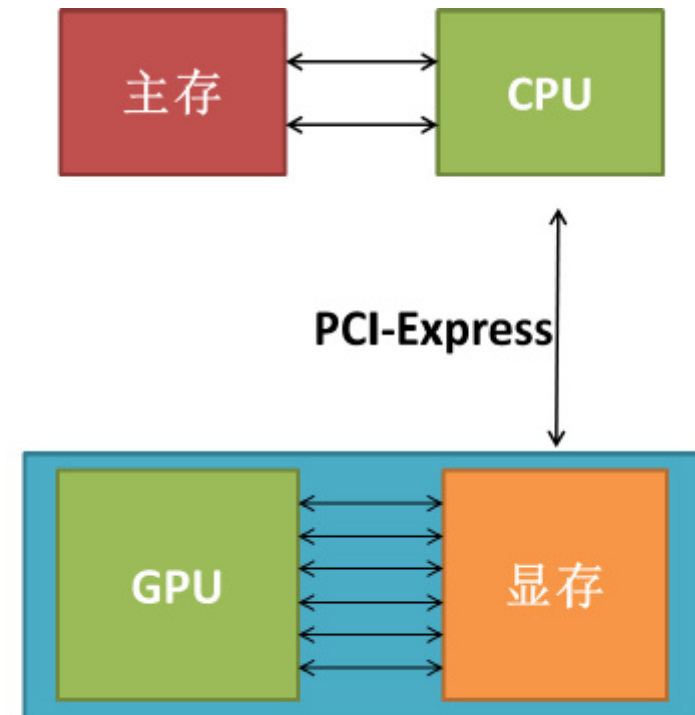
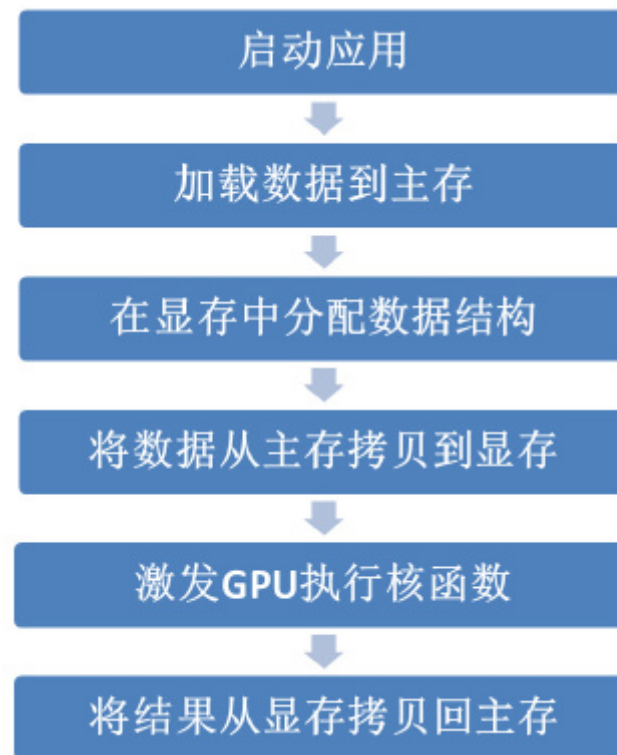


图4 CUDA程序流程

## 5 实践一以图像处理为例

假设我们有如下图像处理任务，给每个像素值加1。并行方式很简单，为每个像素开一个GPU线程，由其进行加1操作。



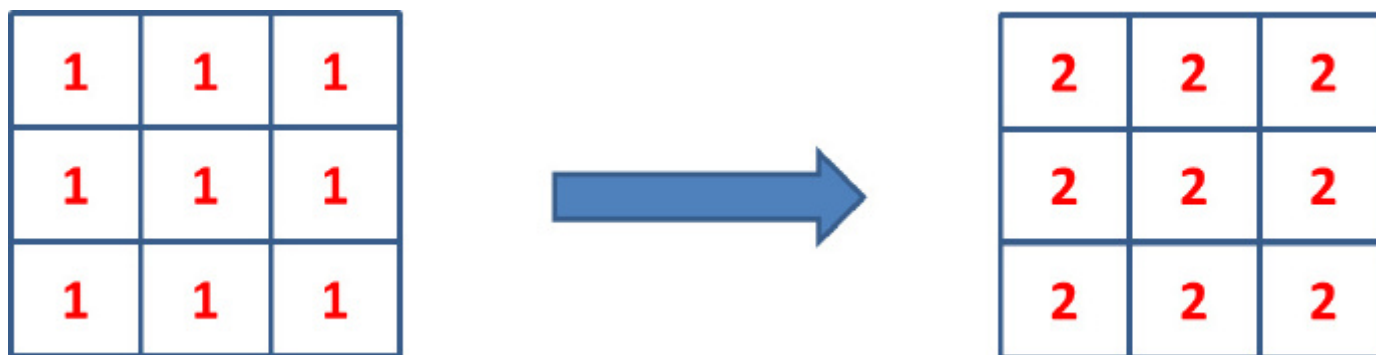


图5 例子

```
//核函数：图像像素值加1
__global__ void cudaImageTransfer(float* gpu_image, int xSize, int ySize)
{
    int i = blockIdx.y*blockDim.y + threadIdx.y;
    int j = blockIdx.x*blockDim.x + threadIdx.x;
    if (i < ySize && j < xSize)
    {
        gpu_image[i*xSize + j] = gpu_image[i*xSize + j] + 1.0;
    }
}
```

图6 核函数

```

int bufSize = xSize * ySize;
//在显存上分配数组
cudaMalloc((void**)&gpu_image, bufSize*sizeof(float));
//将数据从内存传输到显存
cudaMemcpy(gpu_image, cpu_input_image, bufSize*sizeof(float), cudaMemcpyHostToDevice);
//确定线程结构
int perblockSize = 16;
int threadCells = 1;
dim3 threadsPerBlock(perblockSize,perblockSize);
dim3 blocksPerGrid( (xSize+perblockSize*threadCells-1)/perblockSize*threadCells,
(ySize+perblockSize*threadCells-1)/perblockSize*threadCells );
int num = 100;
while(--num > 0) // 迭代100次
{
    //执行核函数
    cudaImageTransfer<<<blocksPerGrid, threadsPerBlock>>>(gpu_image, xSize, ySize);
}
cudaThreadSynchronize();
//结果从显存输出到内存
cudaMemcpy(cpu_output_image, gpu_image, bufSize*sizeof(float), cudaMemcpyDeviceToHost);

```

图7 主流程函数

## 6 GPU加速效果

下图是我实现的基于CUDA的P&D DEM图像预处理算法使用GPU的加速效果，GeForce GT 330是块普通台式机上的显卡，现在价格也就500人民币左右，用它达到了20倍的加速比，Tesla M2075是比较专业的显卡，价格一万左右，用它达到了将近百倍的加速比，这个程序i7 CPU单进程单线程要跑2个小时，而用Tesla M2075 GPU只花了一分多钟就完成计算。



## 加速比（串行计算时间/GPU计算时间）

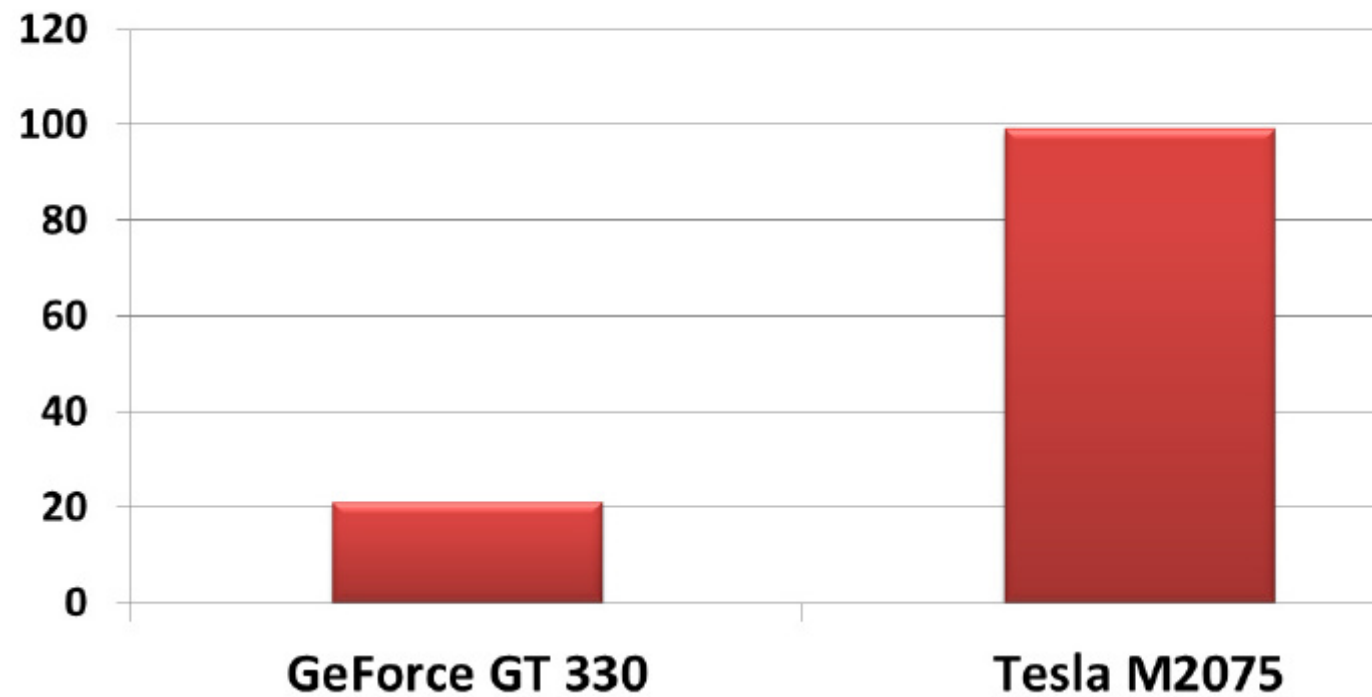


图8 P&D DEM图像预处理算法加速效果

附录（发表的GPU算法相关的SCI论文）：多流向算法GPU并行化

标签：并行计算

好文要顶

关注我

收藏该文



mydddfly

关注 - 12

粉丝 - 11

+加关注

0

0

« 上一篇：双摄像头测距的OpenCV实现

» 下一篇：Android平台Camera实时滤镜实现方法探讨(三)--通过Shader实现YUV转换RGB

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。**

【推荐】超50万VC++源码：大型工控、组态\仿真、建模CAD源码2018！

【推荐】怎样购买腾讯云服务器更划算？



#### 最新IT新闻：

- 张旭豪：创业是顺势而为，要永葆激情
  - 阿里云国际化落地首秀：希望把杭州城市大脑的成功搬到马来西亚
  - 提高份额！微信支付允许用户绑定境外信用卡
  - Microsoft Launcher更新：新增Trending on Bing新功能
  - OPPO可折叠手机专利曝光 手机最终形态
- » 更多新闻...



#### 最新知识库文章：

- 领域驱动设计在互联网业务开发中的实践
- 步入云计算

- 以操作系统的角度述说线程与进程
- 软件测试转型之路
- 门内门外看招聘
- » 更多知识库文章...

#### **历史上的今天:**

2016-06-08 BilibiliDownload

2016-06-08 bilibili(哔哩哔哩) 视频源地址获取原理及源码

Copyright ©2018 mydddfly