关于 GPU 加速的技术

GPU 加速技术应用领域主要有两块，一是 Tensorflow on spark,二是在 PARANSQL（基于 POSTGREL），采用了 PG-STORM 的 GPU 加速， 在基于第二代文件系统和大数据的系统架构下，全面支持 GPU 的加速。

- GPU 的选择

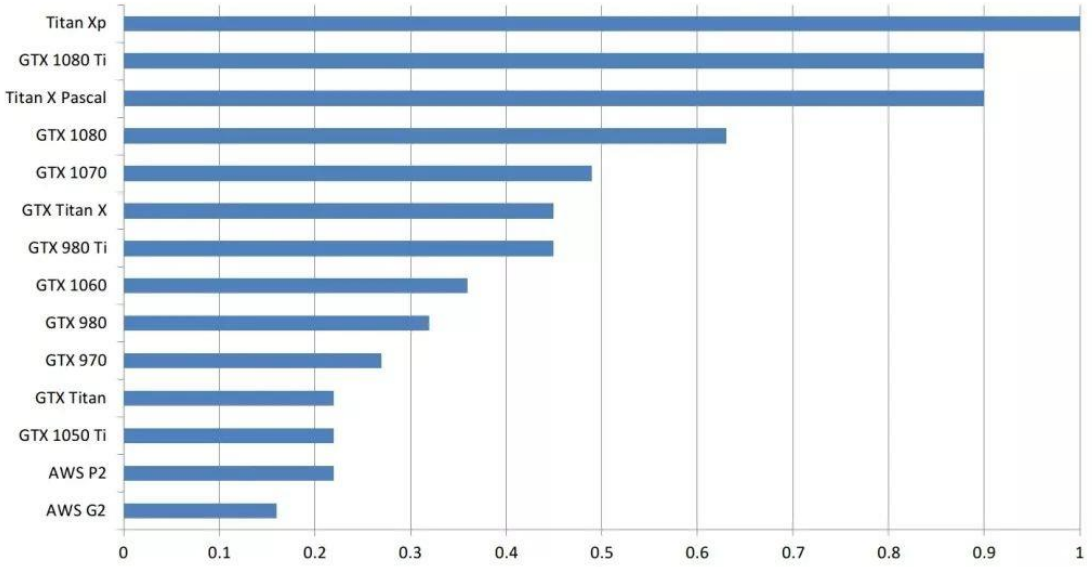
加速的方式有 GPU,TPU（专用 ASIC，如谷歌）和 FPGA，目前通常的选择还是 GPU。

NVIDIA 的标准库使得在 CUDA 中建立第一个深度学习库非常容易，而 AMD 的 OpenCL 则没有这样强大的标准库。现在，AMD 卡没有很好的深度学习库，GPU 计算或 GPGPU 社区对于 CUDA 来说是非常大的，而对于 OpenCL 而言是相当小的。

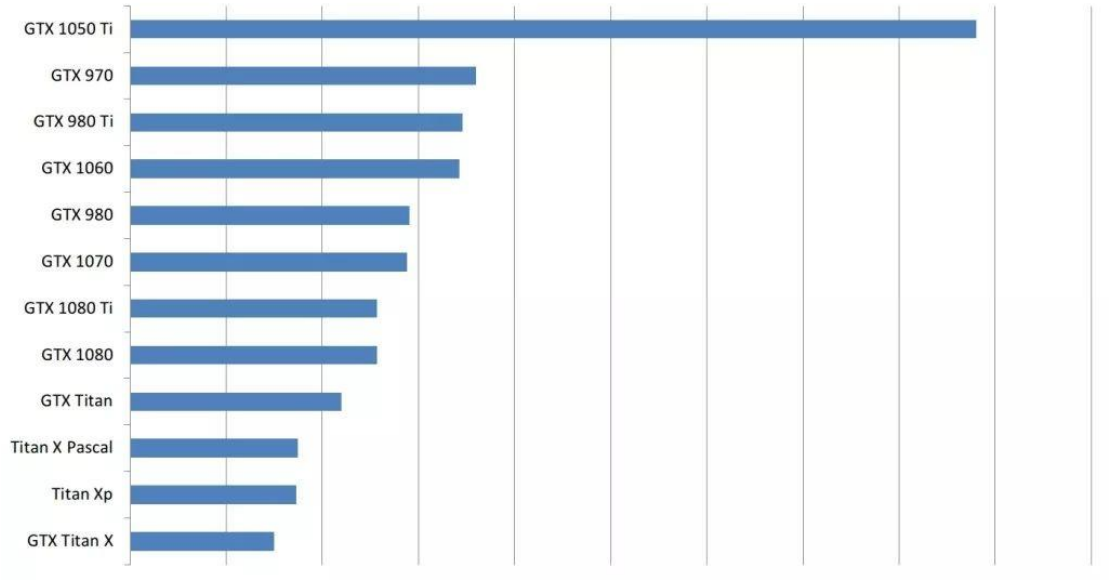
在英特尔至强融核的情况下，广告宣称您可以使用标准的 C 代码，并将代码轻松转换为加速 Xeon Phi 代码。但是，实际上只有很小部分的 C 代码是被支持的，所以这个功能并不是很有用，而且你可以运行的大部分 C 代码都很慢。

曾经在一个至少有 500 个至强 Phis 的 Xeon Phi 集群上工作，因为 Xeon Phi MKL 与 Python Numpy 不兼容；不得不重构大部分代码，因为英特尔至强融核编译器无法对模板进行适当的缩减，而且 Intel Xeon Phi 编译器不支持一些 C ++ 11 功能。

GPU 针对内存带宽进行了优化，同时牺牲了内存访问时间（延迟）。CPU 的设计恰恰相反：如果涉及少量内存（例如乘以几个数字（3 * 6 * 9）），CPU 可以快速计算，但是对于大量内存（如矩阵乘法（A * B * C）他们很慢。由于内存带宽的限制，图形处理器擅长涉及大量内存的问题。当然，GPU 和 CPU 之间还有更复杂的区别



GPU 之间粗略的性能比较。此比较仅适用于较大的工作负载

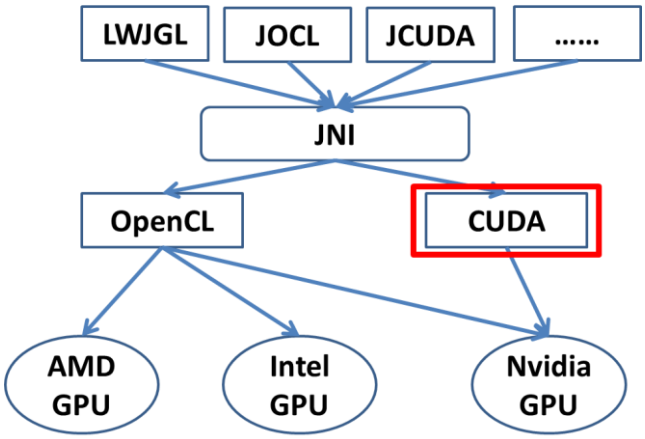


使用上面粗略的性能度量标准和亚马逊的价格来计算新卡的成本效率和旧卡的 eBay 价格。

最好的 GPU 整体（小幅度）：Titan Xp
成本效益，但昂贵：GTX 1080 Ti，GTX 1070，GTX 1080
成本效益和便宜：GTX 1060（6GB）

● GPU 的软件支持库

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 OpenCL，只要有一套 OpenCL 的编程库就能对各类型的 GPU 芯片适用。OpenCL 做到通用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。



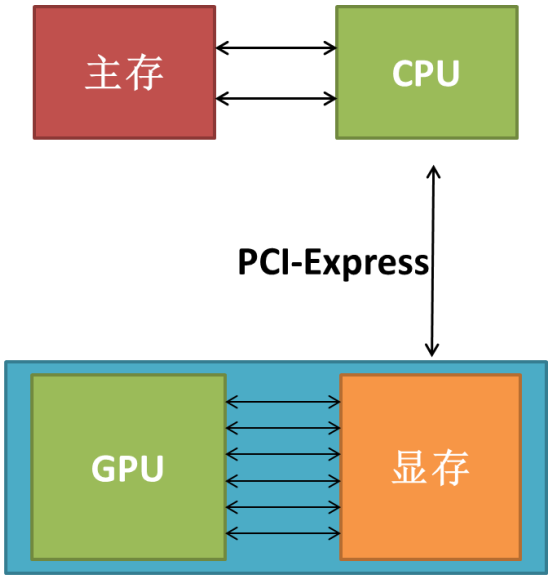
LWJGL (http://www.lwjgl.org/)

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

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

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

JavaCL (http://code.google.com/p/javacl/)



- GPU 在 OALP 分析算法上的应用

主流 GPU 数据库

SQream DB

ZILLIZ（上海）

MapD

Kinetica

BlazingDB

Blazegraph

PG-Strom

Uber AresDB：

● PG-Storm 平台

PG-Strom 是为 PostgreSQLv9.5 或更高版本数据库设计的一个扩展模块。它被设计成利用 GPU 处理单元的并行计算能力，对海量数据执行查询及处理。它的基本理念是 CPU 和 GPU 利用自身优势专注于工作量，并同时执行。CPU 有很多的灵活性，因此，它的优势是操作复杂的东西，如 Disk I / O，另一方面，GPU 在并行数值计算方面有优势，因此，它擅长做大规模但简单的事情，比如多表 JOIN。

在查询优化阶段，PG-Storm 检测给定查询是否完全或部分可以在 GPU 上执行，而后确定该查询是否可转移。如果该查询可以转移，那么 PG-Strom 则在运行中创建 GPU 本地二进制文件的源代码，在执行阶段前启动即时编译进程。接下来，PG-Strom 将提取行集装载入 DMA 缓存（一个缓存区的大小默认为 15MB），并异步启动 DMA 传输和 GPU 内核执行。CUDA 平台允许这些任务在后台执行，因此 PostgreSQL 可以提前运行当前进程。通过 GPU 加速，这些异步相关切分也隐藏了一般延迟。

Hardware Validation List

SuperServer SYS-1019GP-TT

Component List:

Type	Parts Name	# of items	Memo
Barebone	SYS-1019GP-TT	1	1U Rack Server
CPU	Intel Xeon Gold 6126T	1	12C,24HT,2.6GHz
RAM	DDR4-2666 32GB DIMM	6	192GB in total
GPU	NVIDIA Tesla P40	1	3840C, 24GB, Pascal
SSD	Intel DC P4600	3	2.0TB, HHHL
HDD	SATA 2.0TB (2.5inch; 7.2krpm)	6	
N/W	Built-in 10Gb ethernet	2	

Installation Note

- NVIDIA Tesla P40 GPU is installed on the left PCIe slot (x16) from the rear. (https://github.com/heterodb/pg-strom/wiki/001:-GPU-Availability-Matrix)
- Intel DC P4600 SSDs are installed on the center PCIe slot (x16), right upper/lower PCIe slot (x8) (by riser card) from the rear.

Software Support List

- Operating System
  - CentOS 7.3 or later (x86_64)
- CUDA Toolkit
  - NVIDIA CUDA Toolkit 10.0

- PostgreSQL
  - PostgreSQL v9.6.x
  - PostgreSQL v10.x

This section shows a list of NVMe-SSD systems which we have tested in the past.

Vendor	Model	Form	Capacity	SeqRead [MB/s]	PCIe Spec	Memo
Intel	750 SSD	HHHL	400GB	2200MB/s	PCIe 3.0 x4	
Samsung	PRO 960SSD	M.2	512GB	3500MB/s	PCIe 3.0 x4	using carrier board
HGST	SN260	HHHL	7.6TB	6100MB/s	PCIe 3.0 x8	
Intel	DC P4600	HHHL	2.0TB	3200MB/s	PCIe 3.0 x4	
Intel	DC P4511	M.2	1.0TB	1950MB/s	PCIe 3.0 x4	using carrier board

Post OS Installation Configuration

- Setup EPEL Repository
- HeteroDB-SWDC Installation
- CUDA Toolkit Installation
- PostgreSQL Installation(HAWQ installation)
- PG-Strom Installation
- NVME-Strom module(supporting SSD to GPU)

- GPU 在 TensorFlow on spark 上的应用

和深度学习相关的主要 GPU 性能指标如下：

- 内存带宽：GPU 处理大量数据的能力，是最重要的性能指标。
- 处理能力：表示 GPU 处理数据的速度，我们将其量化为 CUDA 核心数量和每一个核心的频率的乘积。
- 显存大小：一次性加载到显卡上的数据量。运行计算机视觉模型时，显存越大越好，特别是如果你想参加 CV Kaggle 竞赛的话。对于自然语言处理和数据分类，显存没有那么重要。

英伟达已经关注深度学习有一段时间，并取得了领先优势。他们的 CUDA 工具包具备扎实的技术水平，可用于所有主要的深度学习框架——TensorFlow、PyTorch、Caffe、CNTK 等。但截至目前，这些框架都不能在 OpenCL（运行于 AMD GPU）上工作。由于市面上的 AMD GPU 便宜得多，希望这些框架对 OpenCL 的支持能尽快实现。而且，一些 AMD 卡还支持半精度计算，从而能将性能和显存大小加倍。

GPU 还需要以下这些硬件才能正常运行：

- 硬盘：首先需要从硬盘读取数据，我推荐使用固态硬盘，但机械硬盘也可以。

- CPU：深度学习任务有时需要用 CPU 解码数据（例如，jpeg 图片）。幸运的是，任何中端现代处理器都能做得不错。
- 主板：数据需要通过主板传输到 GPU 上。单显卡可以使用几乎任何芯片组都可以使用。
- RAM：一般推荐内存的大小至少和显存一样大，但有更多的内存确实在某些场景是非常有帮助的，例如我们希望将整个数据集保存在内存中。
- 电源：一般来说我们需要为 CPU 和 GPU 提供足够的电源，至少需要超过额定功率 100 瓦。