**华 中 科 技 大 学**

**研究生课程考试答题本**

**考生姓名**

**考生学号**

**系、年级** 武汉光电国家实验室 2017级

**类 别**  非定向

**考试科目** 高级计算机系统结构

**考试日期** 2017 **年** 12 **月** 18 **日**

评 分

|  |  |  |  |
| --- | --- | --- | --- |
| 题 号 | 得 分 | 题 号 | 得 分 |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| 总 分： | 评 卷 人： |

注：1、无评卷人签名试卷无效。

2、必须用钢笔或圆珠笔阅卷，使用红色。用铅笔阅卷无效。

# 《张量处理单元的在数据中心中的性能分析》论文阅读报告

# 论文主要内容

## 摘要(Abstract)

许多架构师认为，成本-能源性能的重大改进现在必须来自领域特定的硬件。本文评估自2015年以来在数据中心部署的用来加速神经网络（NN）的推理阶段的定制ASIC（称为张量处理单元（TPU））。 TPU的核心是一个65,536的8位MAC矩阵乘法单元，可提供92 TeraOps /秒（TOPS）的峰值吞吐量和一个大型（28 MiB）软件管理的片上存储器。 TPU的确定性执行模型相比于CPU和GPU的随时间优化方法（相比于保证的latency更有助于平均的吞吐量）能更好的匹配Google的NN应用程序的99％响应时间要求。缺乏这样的功能有助于解释为什么尽管有无数MAC和大内存，TPU相对较小，功耗较低。作者们将TPU与服务器级别的Intel Haswell CPU和Nvidia K80 GPU进行了比较，这些GPU是同一时间部署在同一数据中心的GPU。Google的工作负载是用高级TensorFlow框架编写的，用生产NN应用程序（MLP，CNN和LSTM），这些应用程序占Google数据中心NN推理需求的95％。尽管在一些应用中使用率较低，TPU的平均速度比现在的GPU或CPU要快15倍~30倍，而TOPS/Watt约高出30倍~80倍。而且，在TPU中使用GPU的GDDR5内存将达到TOPS的三倍，并将TOPS/Watt提高到GPU的70倍和CPU的200倍。

## 介绍神经网络(Introduction To Neural Networks)

云中的大型数据集与众多支持它的计算机之间的协同作用使得机器学习得以复兴。特别是深度神经网络（DNNs）已经取得了突破性的进展。DNN的“深”部分来自于它的层数超越了简单的几层，因为云中的大数据集允许使用更大和更多的层捕获更高层次的模式或概念来构建更精确的模型，并且GPU提供足够的计算发展他们。

神经网络的两个阶段被称为训练（或学习）和推理（或预测），它们是指发展与生产。 开发人员选择层数和NN的类型，训练确定权重。 几乎今天所有的培训都是浮点的，这也是GPU如此受欢迎的原因之一。 称为量化的步骤将浮点数转换为窄整数（通常只有8位），这通常足以进行推理。 与IEEE 754 16位浮点乘法相比，8位整数乘法可以减少6倍的能耗，减少6倍的面积，整数加法的优势是在能耗上是13倍，面积是38倍。

目前流行的神经网络有三种：1.多层感知器（MLP）2.卷积神经网络（CNN）3.递归神经网络（RNN）。 最流行的RNN是长期短期记忆（LSTM）。这三种神经网络代表Google数据中心的神经网络推理工作量的95％，作者们用它作为基准。这篇paper的基准测试是在主机服务器上运行的小型应用程序，可以是数千到数百万行的C ++代码。 应用程序通常面向用户，这导致了严格的响应时间限制。

每个模型需要5M到100M的权重，这会花费大量的时间和精力来访问。为了分摊访问成本，在推理或训练期间，在一批独立的例子中重复使用相同的权重，这提高了性能。

本文介绍和测量张量处理单元（TPU），并将其性能和推断能力与当前的CPU和GPU进行比较。 以下是亮点的预览：

•推断应用程序通常强调吞吐量的响应时间，因为它们通常是面向用户的。

•由于延迟限制，尽管K80 GPU具有更高的峰值性能和内存带宽，但推理速度比Haswell CPU快一点。

•尽管大多数架构师正在加速CNN，但他们只占Google数据中心工作量的5％。 TPU的推理速度比K80 GPU和Haswell CPU快15倍~30倍。

•六个NN应用程序中有四个是内存绑定的; 如果TPU被修改为具有与K80 GPU相同的内存，则将比GPU和CPU快30-50倍。

•尽管芯片体积更小，功耗更低，但TPU的内存容量是K80 GPU的25倍，片上内存的3.5倍。

•TPU的性能/功耗是当代CPU和GPU的30倍至80倍; 一个带有K80内存的修改过的TPU将会性能功耗比会是CPU和GPU的70 – 200倍。

## TPU起源，架构，实施和软件(TPU Origin,Architecture,Implementation,And Software)

2013年时，随着随着语音识别，语音搜索的发展，DNN的对数据中心的计算

需求提升了一倍，这使得使用传统的CPU将会非常昂贵。因此开始快速生成一个定制的ASIC的项目，目标是在GPU上提高10倍的性价比。根据这一要求，在短短的15个月内，TPU被设计，验证，建立和部署在数据中心。

为了减少延迟部署的可能性，TPU被设计成PCIe I/O总线上的一个协处理器，允许它像GPU一样插入现有的服务器。而且，为了简化硬件设计和调试，主服务器发送TPU指令来执行，而不是由TPU自行获取。因此，TPU与FPU（浮点单元）协处理器在精神上更接近于GPU。目标是在TPU中运行整个推理模型，以减少与主机CPU的交互，并足够灵活地匹配2015年及以后的NN需求。

文章介绍了TPU的架构。介绍了TPU的功能部件（PCIe Gen3 x16 bus、instruction buffer、Matrix Multiply Unit、Accumulators、Weight FIFO、Weight Memory、Unified Buffer）及其工作原理。文章展示了TPU模具的平面图。 24 MiBUnified Buffer几乎是裸片的三分之一，矩阵乘法单元是四分之一，因此数据路径几乎是裸片的三分之二。控制器只占2％。TPU可像SATA磁盘那样插入现有的服务器。

下面是TPU指令集的情况。当指令通过相对较慢的PCIe总线发送时，TPU指令遵循CISC传统，包括重复字段。这些CISC指令的每条指令（CPI）的平均时钟周期一般为10到20个。它总共有十几条指令，但是Read\_Host\_Memory、Read\_Weights、MatrixMultiply / Convolve、Activate、Write\_Host\_Memory这五条指令是关键的。TPU微架构的理念是保持矩阵单元繁忙。对于这些CISC指令，它使用了一个4级流水线，每个指令在不同的阶段执行。

TPU软件栈必须与为CPU和GPU开发的软件栈兼容，以便应用程序能够快速移植到TPU。在TPU上运行的应用程序部分通常是用TensorFlow编写的，并被编译成可以在GPU或TPU上运行的API 。像GPU一样，TPU堆栈被分成一个用户空间驱动器和一个内核驱动器。内核驱动程序是轻量级的，只处理内存管理和中断。它被设计为长期稳定。用户空间驱动程序经常更改。它设置并控制TPU执行，将数据重新格式化为TPU顺序，将API调用转换为TPU指令，并将其转换为应用程序二进制文件。用户空间驱动程序在首次评估时编译模型，缓存程序图像并将重量图像写入TPU的权重内存;第二次和以后的评估全速运行。 TPU将大部分模型从输入到输出完全运行，从而使TPU计算时间与I / O时间之比最大化。计算通常一次完成一层，重叠执行允许矩阵乘法单元隐藏大多数非关键路径操作。

## CPU, GPU, 和TPU 平台(CPU,GPT,And TPU Platforms)

工作负载是六个神经网络的程序，这六个程序代表了Google数据中心95％的TPU使用量。

基准平台是服务器级计算机，在2015年部署TPU时可用。 这个限制意味着它们必须至少包括内部SRAM的SECDED保护以及像TPU这样的外部DRAM存储器，不包括诸如Nvidia Maxwell GPU之类的选择。对于Google公司来购买和部署它们，它们也必须是明智的配置机器，而不是为了赢得基准测试而组装的尴尬的文物。

如下图所示，原文中Table 2 列出了所用于测试的服务器。



原文表2列出了Google的选择。 传统的CPU服务器由Intel的18核双插槽Haswell处理器代表。 该平台也是GPU或TPU的主机服务器。 Haswell在英特尔的22纳米工艺中被制造出来。 CPU和GPU都是非常大的裸片：约600平方毫米！

2.3 GHz的CPU时钟频率不包括Turbo模式，因为它很少发生在NN应用程序的数据中心。 Haswell具有不同的时钟频率，具体取决于程序是否使用Google的NN应用程序经常使用的AVX指令。 Turbo模式的更高时钟频率（避免AVX的程序）在不使用所有内核时发生。 因此，在Google的数据中心中，Turbo模式很少见的另一个原因是Google的应用程序通常使用所有内核，另外还可以运行其他数据中心作业来填充所有空闲内核。

GPU加速器是Nvidia K80。每个K80卡包含两个管芯，并在内部存储器和DRAM上提供SECDED。 Nvidia表示，“K80加速器通过使用更少，更强大的服务器提供应用程序性能，大大降低了数据中心成本”。 神经网络的研究人员在2015年经常使用K80，最近在2016年9月被选中用于新的基于云的GPU产品。 该服务器上最多可以安装八个K80模具，这是Google基准测试的配置。

由于每个服务器的die的数量在2到8之间变化，原文通常显示每个die标准化的结果（原文图5-8，图10-11和表3,5和7），但是原文偶尔会显示整个系统（图9）。

## 性能: ROOFLINES, 响应时间和吞吐量(Performance: Rooflines, Responsetime,And Throughput)

为了说明三个处理器上六个应用程序的性能，原文调整了来自高性能计算（HPC）的Roofline性能模型。 这个简单的视觉模型并不完美，但它提供了洞察性能瓶颈的原因。 模型背后的假设是应用程序不适合片上高速缓存，所以它们要么是计算受限的，要么是内存带宽受限的。 对于HPC来说，Y轴是每秒浮点运算的性能，因此峰值计算速率形成了车顶线的“平坦”部分。 X轴是操作强度，按访问每个DRAM字节的浮点操作度量。 内存带宽是每秒字节数，因为（FLOPS / sec）/（FLOPS / Byte）= Bytes / sec，所以成为屋顶线的“倾斜”部分。 如果没有足够的操作强度，程序会受到内存带宽的限制，并且会处于车顶线的倾斜部分。

应用程序的每秒实际操作与其正上方的上限之间的差距显示进一步性能调整的潜在益处，同时保持操作强度不变; 当然，增加运营强度的优化（如缓存阻塞）可能会带来更大的收益。

要为TPU使用Roofline模型，当量化NN应用程序时，作者们首先用整数运算替换浮点运算。 由于权重通常不适合NN应用的片上存储器，所以第二个改变是将操作强度重新定义为读取权重的每个字节的整数运算。



原文图5显示了对数标尺上单个TPU die的车顶线模型。TPU在车顶线上有一长长的“倾斜”部分，操作强度意味着性能受内存带宽的限制，而不是由计算的峰值性能所限制。六个应用程序中有五个正在高高兴兴地与天花板碰撞：MLP和LSTM是内存限制的，CNN是计算限制的。尽管运行强度很高，CNN1只有14.1TPS，而CNN0在86TPS上运行。



原文表3解释了CNN1发生的情况，根据性能计数器给出了TPU操作的部分可见性。 TPU花费的时间少于执行CNN1矩阵操作的一半周期（第7列，第1行）。 在每个有效周期中，只有65,536个MAC中的一半保持有用的权重，因为CNN1中的一些层具有浅的特征深度。 大约35％的周期花在等待权重从内存加载到矩阵单元中，这发生在4个完全连接的层中，运行强度只有32（见第8节的最后一个谬误）。 这大约有19％的周期没有被矩阵相关计数器解释。 由于在TPU上重复执行，作者们没有精确计算这些周期，但是可以看到23％的周期在管道中有RAW依赖性的停顿，1％的停机时间用于输入PCIe总线。



原文图6和图7显示了单个Haswell die和单个K80 die的顶线。 六个NN应用程序通常比图5中的TPU更低。响应时间是原因。 这些NN应用程序中有许多是面向终端用户的服务的一部分。 研究人员已经证明，响应时间的小幅增加会导致客户使用更少的服务。 因此，虽然培训可能没有硬性的回应时间最后期限，推理通常会。 也就是说，推断相对于吞吐量来说更偏好延迟。



例如，MLP0的99百分位响应时间限制为7 ms，这是应用程序开发人员所要求的。（每秒推断和7毫秒延迟包括服务器主机时间以及加速时间。）原文表4显示，如果响应时间限制被放宽，Haswell和K80分别以MLP0可达到的最高吞吐量的42％和37％运行。 这些边界也会影响TPU，但是在80％的情况下它的运行更接近MLP0的最高吞吐量。 与CPU和GPU相比，单线程TPU没有复杂的微架构功能，消耗晶体管和能量来改善平均情况，但不是99％的情况：没有高速缓存，分支预测，无序执行，多处理，推测性预取，地址合并，多线程，上下文切换等等。 极简主义是领域特定处理器的优点。

原文表3显示了TPU的性能，但是它没有考虑主机服务时间，可以分为运行应用程序的主机共享和与TPU通信。原文表5列出了第二部分，但第一部分很难。 排队论表明，长输入队列通过确保计算机从不空闲来提高吞吐量，但延长了响应时间。 因此，大多数应用程序保持输入队列为空。 作者们无法测量TPU何时处于空闲状态，因为它正在等待CPU执行其部分应用程序，或者由于空的输入队列而导致CPU也处于空闲状态。



表6给出了每个die相对推断性能的底线，包括两个加速器相对于CPU的主机服务器开销。 倒数第二列显示了六个NN应用的相对性能的几何平均值，这表明K80芯片是Haswell芯片速度的1.1倍，TPU芯片的速度是14.5倍，因此TPU die速度是GPU die的13.2倍。 图8直观地显示了它们的相对速度。





当架构师不知道将要运行的程序的实际组合时，使用几何平均值。但是，对于这项研究，作者们知道知道这个组合（表1）。 表6最后一列中使用实际混合的加权平均值将GPU增加到1.9倍，TPU增加到29.2倍，所以TPU die现在是GPU die的15.3倍。



## 成本 - 性能，TCO和性能/瓦特(Cost-Performance,TCO, And Performance /Watt)

当购买成千上万的电脑时，性价比胜过性能。数据中心的最佳成本指标是总体拥有成本（TCO）。Google为数千芯片支付的实际价格取决于相关公司之间的谈判。出于商业原因，这篇paper不能发布这样的可能会被推断出来的价格信息或数据。然而，功耗与TCO相关，并且文章中可以发布每个服务器的功耗，因此文章中使用性能/功耗作为性能/ TCO的代理。在本节中，作者们比较了整个服务器，而不是单个的die。

图9显示了K80 GPU和TPU相对于Haswell CPU的几何和加权平均性能/瓦数。作者们提出两个不同的性能/瓦特计算。第一个（“总计”）包括主CPU服务器在计算GPU和TPU的性能/功耗时所消耗的功耗。第二个（“增量”）预先从GPU和TPU中减去主机CPU服务器的功耗。对于总体性能/瓦特，K80服务器是Haswell 的1.2-2.1倍。对于增量性能/瓦特，当Haswell服务器电源被省略时，K80服务器是CPU的1.7-2.9倍。TPU服务器的总性能/功耗比Haswell高出17- 34倍，使得TPU服务器的性能/功耗达到了K80服务器的14-16倍。

## 能耗比例(Energy Proportionality)

散热设计功耗（TDP）会影响供电的成本，因为在硬件满功率时您必须提供足够的电力和散热。然而，电力成本是根据工作量在一天中的变化而平均消耗的。 研究发现，服务器百分之百忙于不到10％的时间，并提倡能源比例：服务器消耗功率应与工作量成正比。上一节中的功耗估算是基于Google数据中心所见的TDP的一部分。

随着提供的工作负载利用率在0％到100％之间变化，论文中测量了包括CPU，TPU和GPU在内的服务器的性能和功耗，并以10％的工作量增量的桶形式收集。图10显示了通过改变CNN0的工作负载，服务器功率除以每个服务器的裸片数量。

文中发现TPU具有最低的功率（每芯片40W），但能量比例较差：在10％的负载下，TPU使用100％的功率的88％（这个简短的设计方案阻止了很多节能功能的出现）。Haswell在能源比例上是最好的：它在10％的负载下使用了56％的功率。 K80比TPU更接近CPU，在10％工作负载下使用66％的满载功率。不受计算限制的LSTM1表现也是类似：在10％的负载下，CPU使用全功率的47％，GPU使用78％，而TPU使用94％。

如果CNN0成为加速器的主机，运行CNN0时服务器的能耗使用情况如何？ 当GPU和TPU处于100％负载时，CPU服务器使用GPU的52％的全功率和TPU的69％（因为CPU的运行速度比GPU快得多，所以CPU运行得更多）。因此，Haswell服务器加上四个TPU的附加功耗<20％，但运行CNN0的速度比单独的Haswell服务器快80倍（4个TPU vs 2个CPU）。

## 对TPU替代设计的评估(Evaluation Of Alternative TPU Designs)

与FPU一样，TPU协处理器也相对容易评估，因此文章中为六个应用程序创建了一个性能模型。表7显示了模型结果和硬件性能计数器之间的差异，平均低于10％。然后，作者们通过改变内存带宽，时钟速率和累加器数量以及矩阵乘法单元大小来模拟性能。



图11显示了随着将这些参数在0.25至4倍范围内进行缩放的TPU芯片的平均性能灵敏度。它绘制的是加权平均值，但几何平均值看起来相似。除了评估仅提高时钟速率（图11中的时钟）的影响之外，作者们还绘制了一个设计（clock+），其中时钟速率增加并且累加器的数量相应地缩放，以便编译器可以在飞行中保持更多的存储器参考。同样，如果增加一维（matrix+）的上升平方的累加器数量，绘制矩阵单元扩展。因为矩阵中乘法器的数目在两个维度上都增长，以及仅仅增加矩阵单元（矩阵）。



首先，增加内存带宽的影响最大：当内存增加4倍时，性能平均提高3倍。其次，时钟频率在有或没有更多的累加器的情况下平均没有什么好处。原因是MLP和LSTM是内存受限的，但只有CNN是计算受限的。虽然在图11中很难看出，但是由于它只显示了所有六个DNN的权重的平均值，所以将时钟速率提高4倍对MLP和LSTM几乎没有影响，但是将CNN的性能提高了约2倍。第三，当矩阵单元所有应用程序从256x256扩展到512x512时，图11中的平均性能会略微下降，不管它们是否获得更多的累加器。这个问题类似于大型页面的内部碎片，只是因为它在两个维度上更糟。考虑在LSTM1中使用的600x600矩阵。使用256x256的矩阵单元，需要9个步骤才能平铺600x600，总共需要18us的时间。较大的512x512单位只需要四个步骤，但每个步骤需要四倍的时间，为32 us的时间。TPU的CISC指令很长，所以解码是不重要的，不会掩盖从DRAM加载的开销。



表8显示了24 MiB统一缓冲区的使用情况，最初的规模是允许MLP以最大2048的批量运行。最近，作者们改进了统一缓冲区的存储分配器，减少六个中最大的存储器申请到14 MiB。在开始的18个月内，TPU使用了全部容量，而新的分配器正在开发中。现在额外的容量为采用更大型号增加了余量。

文章接下来使用性能模型来评估假设TPU die。更积极的逻辑综合和块设计可能已经将时钟频率提高了50％。如同K80一样，为GDDR5存储器设计一个接口电路可以将权重存储器的带宽提高五倍以上，从而将其roofline ridge point从1350平移到250. 如图11所示，时钟频率提高到1050 MHz，内存变化不大。如果作者们将时钟保持在700 MHz，但将GDDR5用于“weighted memory”，则几何平均增加量将跳至2.6，加权平均值将升至3.9。这两者都提高了几何平均值（2.9），但不是加权平均值，所以TPU只有更快的memory。

图11不包含主机服务时间。作者们使用表5来计算TPU的主机服务器交互开销的时间。加上相同的额外时间降低了TPU' means是从2.6到1.9和从3.9到3.2。这个改变是乐观的，因为它不包括CPU时间来运行其应用程序的份额，但是也是悲观，因为作者们可能会主动调整主机代码给予TPU 3倍的速度。

 用和K80相当的GDDR5存储器替换DDR3 weight memory 需要将存储器通道的数量翻倍为四倍。 这一改进将使芯片尺寸扩大10％左右。 但是，更高的内存带宽降低了统一缓冲区的压力，因此将统一缓冲区减少到14 MiB可能会减少10％的面积。 GDDR5还将TPU系统功率预算从861瓦增加到约900瓦，因为每个服务器有4个TPU。 图9显示TPU的相对总性能/瓦特/die比Haswell高出31倍 - 86倍，在K80高达25倍 - 41倍。 增量指标比Haswell高出69-196倍，在K80上高出42-68倍。

##  讨论(Discussion)

●谬论：数据中心的NN推理应用程序的吞吐量与响应时间相当。

Google的开发人员对响应时间的要求很高，正如2014年的一些人所建议的那样，批量大小足以使TPU达到最佳性能或延迟要求不会太严格。一个驾驶应用程序是离线图像处理，直觉是如果交互式服务也需要TPU，他们中的大多数将会积累更大的批次。即使2014年一款关注响应时间的应用程序的开发者（LSTM1）也表示2014年的限制为10毫秒，但实际上将其移植到TPU的时间缩短到了7毫秒。许多这样的服务对TPU的意外期望以及对低响应时间的影响和偏好改变了方程，应用程序编写者经常选择等待较大批量积累而减少延迟。幸运的是，TPU具有简单且可重复的执行模式，可以帮助满足交互式服务的响应时间目标，并且峰值吞吐量也非常高，即使小批量处理器的性能也比当前的CPU和GPU高

●谬误：K80 GPU架构与神经网络推理相当吻合。GPU传统上被视为高吞吐量架构，依靠高带宽DRAM和数千个线程来实现其目标。这个观点有助于解释为什么K80在推理上比Haswell快一点，比TPU慢得多。 K80的后续产品肯定会包含优化以提高峰值推理性能，但鉴于其面向吞吐量的架构方法，GPU可能更难以满足严格的延迟限制。和如第7节所示，TPU的改进空间很大，因此不是一个容易的目标。

●陷阱：架构师忽视重要的神经网络任务。 作者们很高兴看到架构社区正在向NN提供支持：在ISCA 2016上，有15％的论文是在NN的硬件加速器上的但是却有9篇论文都看了CNN，只有两篇提到了其他的NN。 CNN比MLP更复杂，在NN竞争中更为突出，这可能解释了它们的吸引力，但它们只占Google数据中心NN工作量的大约5％。 虽然CNN在边缘设备中可能很常见，但是卷积模型的数量还没有赶上数据中心的MLP和LSTM。 作者们希望架构师尽可能地加快MLP和LSTM的速度。

●陷阱：对于NN硬件，每秒推测（IPS）是不准确的摘要性能指标。 研究结果表明，对于NN硬件来说，IPS是一个糟糕的总体性能总结，因为它仅仅是应用中典型推断的复杂性（例如NN层的数量，大小和类型）的倒数。 例如，TPU运行4层MLP1,360,000 IPS，而89层CNN1只有4,700 IPS，所以TPU IPS变化了75倍！ 因此，使用IPS作为单速概要对于NN加速器来说，比MIPS或FLOPS用于常规处理器更具误导性，所以IPS的作用应该更加被贬低。 为了更好地比较NN机器，作者们需要一个高级编写的基准测试套件，将它移植到各种各样的NN体系结构中。 在这样一个基准测试套件中，Fathom是一个很有希望的新尝试。

●谬误：如果启用升压模式，则K80 GPU的结果会更好。 作者们没有使用K80升压模式，而是测量了它对LSTM1的影响。 升压模式将时钟频率从560提高到了1.6到1.675，这使性能提高了1.4倍，但功耗也提高了1.3倍。 性能/瓦的净增益是1.1倍，因此升压模式对LSTM1的影响较小。

●谬误：如果更有效地使用它们或者与更新的版本进行比较，则CPU和GPU的结果将与TPU类似。作者们原来在CPU上只有一个DNN的情况下有8位结果，这是因为AVX2整数支持的有效工作非常重要，好处是~3.5倍。将所有的CPU结果以浮点形式呈现出来，而不是有一个例外，并且以自己的车顶线来表达，这样就没有那么混乱（也没有太多的空间）。如果所有的DNN都有类似的加速，性能/瓦特率将从41 – 83倍下降到12 – 24倍。新的16-nm，1.5GHz，250W P40 GPU可以执行47 Tera 8-bit ops / sec，但在2015年初无法使用，因此与文章中用的三个平台不同。也不知道在刚性时间范围内交付的P40峰值的比例。 （它也不提供内部存储器的SECDED，所以不能在数据中心部署）。如果比较新的芯片，第7节表明，可以将28-nm，0.7GHz，40W TPU的性能提高三倍通过使用K80的GDDR5内存（额外的10W的成本）。

●陷阱：将性能计数器添加为NN硬件的事后考虑。 TPU有106个性能计数器，但是作者们希望更多（见表3）。神经网络加速器的存在的理由是性能，在进化过程中，对发生的事情有很好的直觉还为时过早。

●谬误：经过两年的软件调优，增加TPU性能的唯一途径是硬件升级。如果开发人员和编译器编写者做了更多的工作来将CNN1与TPU硬件相匹配，那么CNN1在TPU上的性能可以得到改善。例如，开发人员可以重新组织应用程序，将卷积图层中的多个短批量聚合为四个完全连接的图层的单个较深的批次（从32到128）。这样一个单层将提高矩阵单元的利用率（见表3）。由于CNN1目前在TPU上的运行速度比CPU快70多倍，CNN1开发人员已经非常高兴，所以目前还不清楚是否或何时进行这种优化。

●陷阱：在设计特定于领域的体系结构时，不了解架构历史。不适用于通用计算的思路对于领域特定的体系结构可能是理想的。对TPU而言，三个重要的架构特征可以追溯到20世纪80年代早期：收缩阵列，解耦访问/执行和CISC指令。第一个是减少了矩阵乘法单元的面积和功率，第二个是在矩阵乘法单元运行时同时获取权重，第三个是利用PCIe总线的有限带宽传输指令。 Historyaware架构师可以拥有竞争优势。

## 相关工作(Related Work)

两篇调查文章记载了定制的NN ASIC至少要返回25年前。例如，CNAPS芯片包含一个由16位乘8位乘法器组成的64个SIMD阵列，多个CNAPS芯片可以用一个序列发生器连接在一起。 Synapse-1系统是基于定制的收缩多积累芯片MA-16，每次执行十六个16位乘法。系统将多个MA-16芯片连接在一起，并具有用于执行激活功能的定制硬件。由T0定制ASIC加速的25个SPERT-II工作站于1995年开始部署，用于神经网络训练和语音识别推理。 40Mhz T0向MIPS指令集架构添加了向量指令。八通道矢量单元可以在每个时钟周期基于8位和16位输入产生多达16个32位算术运算结果，比SPARC-20工作站在推理上快25倍，在训练时快20倍。他们发现16位不足以进行训练，所以他们使用了两个16位字，这使得训练时间增加了一倍。为了克服这个缺点，他们引入了32到1000个数据集的“批量”（批量），以减少更新权重所花费的时间，这比单个词但没有批次训练更快。最近的DIANNAO系列NN架构通过对存储器进行有效的架构支持，最大限度地减少了芯片和外部DRAM的存储器访问

在NN应用程序中出现的访问模式。全部使用16位整数运算，所有设计合成为布局，但没有制造芯片。原始的电脑使用64个16位整数乘法累加单元阵列，片内存储器为44KB，估计为3mm2（65nm），以1GHz运行，功耗为0.5W。这些能量大部分都进​​入DRAM访问权限，所以一个接替大电脑（“大电脑”）的eDRAM包含了36Mb的权重。目标是在多芯片系统中有足够的内存来避免外部DRAM访问。后续的点点数（“通用计算机”）更多地针对支持向量机。另一个分支是针对CNN的“时代脑”（“视觉计算机”），它通过将加速器直接连接到传感器来避免DRAM访问。

卷积引擎也专注于CNN进行图像处理。这种设计部署了64个10位多累加器单元，并定制了一个Tensilica处理器，预计在45纳米的800 MHz下运行。与SIMD处理器相比，它的能耗面积效率要高出8倍到15倍，而在为特定内核设计的定制硬件的2倍到3倍之内。 Fathom的基准测试报告似乎与这篇文章的结果是矛盾的，GPU的运行推理要比CPU快得多。然而，他们的CPU和GPU不是服务器端，CPU只有4个内核，应用程序不使用CPU的AVX指令，并且没有响应时间截止。Catapult是使用可重构性来支持DNN的最广泛部署的例子，这是许多人提出的。他们选择了FPGA上的FPGA来降低功耗，以及延迟敏感型应用程序不能很好地映射到GPU的风险。 FPGA也可以重新使用，例如搜索，压缩和网络接口卡。TPU项目实际上是从FPGA开始的，但是后来放弃了。

与当时的GPU相比，当时的FPGA在性能上并不具有竞争力，TPU的功耗可能要比GPU低很多，而且速度可能要比FPGA和GPU要好得多。虽然2014年首次发布，由于Catapult在2015年与TPU同时向数据中心部署了28-nm Stratix V FPGA，因此Catapult是一个现代TPU。Catapult具有200 MHz时钟，3926个18位MAC，5 MiB片上存储器，11 GB / s存储器带宽，并使用25瓦。 TPU具有700MHz时钟，65,536个8位MAC，28个MiB，34GB / s，并且通常使用40瓦。 Catapult的修改版本在2016年以更大规模部署[9]。 Catapult V1运行CNNs--使用一个收缩矩阵乘法器，速度是2.1 GHz，16核，双插槽服务器的2.3倍[39]。使用Catapult V2的下一代FPGA（14-nm Arria 10），性能可能会提高到7倍，甚至更高的性能可能达到17倍，并且更加小心地进行布局。虽然是苹果和橙子相比，但是目前的TPU芯片的CNNs已经提升了40到70倍，而速度更快的服务器（表2和表6）。也许最大的不同在于，为了获得最佳性能，用户必须使用低级硬件设计语言Verilog 编写长程序，而不是使用高级TensorFlow框架编写短程序。也就是说，可重编程性来自TPU的软件，而不是来自FPGA的固件。

## 总结（CONCLUSION）

浮点性能，当大多数主流工作负载变成了整数运算占主导地位时。作者们观察到每秒推测（IPS）更多的是NN的功能，而不是底层硬件，所以IPS是NN处理器的一个更糟糕的单一性能指标，而MIPS和MFLOPS则是CPU和GPU。作者们还了解到，推理应用程序具有严重的响应时间限制，因为它们通常是面向用户的应用程序的一部分，所以NN架构在处理99个百分点延迟期限时需要表现良好。虽然K80可能在训练方面表现出色，但平均而言，它比Haswell在推断的工作量方面稍微快一点，可能是因为它强调吞吐量而不是延迟;这与Google的推理应用程序的严格响应时间期限相冲突。 TPU芯片利用其在MAC和片上存储器中的优势来运行使用特定领域的TensorFlow框架编写的短程序，其速度是K80 GPU芯片的15倍，从而导致每瓦特优势的性能达到29倍，这与每个总体拥有成本的性能。与Haswell CPU芯片相比，相应的比率是29和83.虽然未来的CPU和GPU肯定会更快地运行推理，但使用大约2015年GPU内存的重新设计的TPU将会快两到三倍，并提升性能/瓦特优势在Haswell上超过K80和200将近70。综上所述，TPU由于大而不是太大的矩阵乘法单元而成功，大量的软件控制片上存储器;运行整个推理模型以降低对主机CPU的依赖的能力;一个单线程，确定性的执行模式，证明是一个很好的匹配到99百分位的响应时间限制;有足够的灵活性来匹配2013年以及2017年的NN;尽管数据路径和内存较大，但是省略了通用功能，使得功耗小，功耗低;量化应用程序使用8位整数;而且应用程序是使用TensorFlow编写的，这使得它们可以很容易地以高性能移植到TPU上，而不必重写，以便在不同的TPU硬件上运行良好。产品之间的数量级差异在计算机体系结构中是罕见的，这可能导致TPU成为一种产品特定领域体系结构的原型。

# 论文内容分析

### 论文主要内容

1. **这篇paper尝试去解决什么问题？**

主要想提高人工智能计算任务，尤其是神经网络的执行效率。本文的TPU主

要解决神经网络运行过程中的推理方面的效率提升的问题。

1. **这篇paper的主要的idea是什么？关键的亮点？**

主要亮点就是设计了TPU，并详细地测试了其性能。

1. **What is the key contribution to literature at the time it was written?**

本文结构清晰，但不是经典的论文写法。Abstract概括文章的内容，然后交代文章的意义和背景。最后交代了TPU的性能的表现。第一节，Introduction介绍了神经网络相关的知识。第二节，交代TPU的起源，架构和软件栈等。第三节介绍了CPU、GPU和TPU对比的平台。第四五六七节详细介绍了不同方面的测试结果。最后是讨论，相关工作和总结等等。

这篇文章的结构实际上和通常的文章结构大体一致，只是重点突出了各种性能的评测，通常的paper只是在实验那一章去讲。这也恰恰说明了这篇文章的重点在各种的性能的评测上面。

1. **从这篇paper中学到了什么最重要的东西？**

对一个处理器的性能的评测方法。

Google自己开发了一款名为TPU的专用芯片，用于TPU 英文全称 Tensor Processing Unit(张量处理器)。是谷歌专为它的深度学习语言TensorFlow 开发的一种芯片。TPU 是一种ASIC,这是一种与FP-GA 类似,但不存在可定制性的专用芯片,经过了专门深度机器学习方面的训练,它有更高效能(每瓦计算能力),宽容度更高,每秒在芯片中可以挤出更多的操作时间,使用更复杂和强大的机器学习模型,将之更快的部署,用户也会更加迅速地获得更智能的结果,在处理人工智能方面功耗更低,最出名的就是AlphaGo。

另外从商业角度讲,目前Google 的处理器还是自产自销,不会对外出售。不过随着越来越多的企业采用Google 云服务,采购Intel 处理器并自建服务器/数据中心的会越来越少,会对Intel、NVIDIA 这样的大客户产生巨大冲击。

本文介绍和测量张量处理单元（TPU），并将其性能和推断能力与当前的CPU和GPU进行比较。

以下是这篇文章的主要内容：

•推断应用程序通常强调吞吐量的响应时间，因为它们通常是面向用户的。

•由于延迟限制，尽管K80 GPU具有更高的峰值性能和内存带宽，但推理速度比Haswell CPU快一点。

•尽管大多数架构师正在加速CNN，但他们只占我们数据中心工作量的5％。 TPU的推理速度比K80 GPU和Haswell CPU快15倍~30倍。

•六个NN应用程序中有四个是内存绑定的; 如果TPU被修改为具有与K80 GPU相同的内存，则将比GPU和CPU快30-50倍。

•尽管芯片体积更小，功耗更低，但TPU的内存容量是K80 GPU的25倍，片上内存的3.5倍。

•TPU的性能/功耗是当代CPU和GPU的30倍至80倍; 一个带有K80内存的修改过的TPU将会是CPU和GPU的70 – 200倍。

### 长处

这篇paper可以说达到了目标，实现了特定场域下的高效率：

TPU 可适用现存各种神经网络模型，从影像辨识的 CNN 到语音识别的

LSTM，都适用。TPU 跟 CPU 或 GPU 一样是程序化的。 TPU 不是专为某一个神经网络模型设计;TPU 能在多种网络(卷积网络、LSTM 模型和大规模全联机的神经网络模型)上执行 CISC 指令。

 Google产品的人工智能负载，主要利用神经网络的推理功能，其 TPU 处理速度比目前 GPU 和 CPU 要快 15 到 30 倍。较之传统芯片，TPU 也更加节能，功耗效率(TOPS/Watt)提升了 30 到 80 倍。驱动这些应用神经网络只要求少量程序代码，少得惊人：仅 100~1,500 行,且程序代码以TensorFlow 为基础。

 效能上看，确实透过裁剪运算器宽度(8 位定点)达到非常高的理论峰值，在大多数卷积作业效率很好。 但在部分其他类别的神经网络作业上，效率不是太高。

 从软硬件结合的角度看，TPU 强大的地方不完全在芯片，而是结合 TensorFlow 的软件生态。

### 短处

需要明确说明的是：TPU 是一款推理芯片，并不是用做训练。

在深度学习领域，主要有 3 种运算方式：（1）数据中心的训练(用于训练模型，计算机往往需要以较高的精确度运算，一般使用 32 位浮点运算)。

（2）数据中心的推理(在云端进行实时连续运算，精确度可以适当牺牲，换取更快的速度和更低能耗)。（3）内嵌式装置的推理(这类应用的核心是低能耗的 ASICs)。所以 TPU 针对的，就是第二种运算方式：数据中心的推理。 对推理运算而言，重要指针就是快速和低能耗。

 TPU 并不是全能，只是用于数据中心的推理阶段。 深度学习模型的前期训练，也还是要依靠高精确度的 GPU。而且 TPU 通用性差。这就是TPU的短板，还有其和Tensorflow的关系太过密切，可能对其他的应用程序来说并没有那么好的效果。

### Can you do (much) better? Present your thoughts/ideas.

因为好多我都没看懂，所以我觉得自己不会比这篇文章做得更好了。

### 从这篇paper中学到了什么？

我觉得这篇文章的实验特别多，数据详实，内容丰富，值得学习，还有就是

水平看起来挺高的，我都不知道它在说什么，还需要学习啊。

# 详细分析两幅图表

1. **论文中的Figure 9**



1. **本实验的目标**

目标是想比较TPU的服务器和装有GPU、CPU的服务器之间的性价比（Cost-

Performance）

1. **本实验的设计思想**

本来衡量性价比的最佳指标是performance/TCO,但是由于商业保密的原因，并不公开价格信息，所以使用performance/Watt来衡量性价比，因为功率和TCO是相关的。

1. **本实验的具体配置**

本实验所用的服务器在原文表二中列出



 本实验所用的工作负载在原文中表三中列出，是一些运行MLP，CNN，LSTM神经网络的应用程序



1. **本实验图中各种标记的含义**

**横坐标**是四种不同情况下的性价比。总体的性价比的几何平均值，总体的性价比的加权平均值，增量式的性价比的几何平均值，增量式的性价比的加权平均值。

**纵坐标**是相对于CPU或GPU的性价比（Performance/Watt）

**图表类型**是一个柱状图.

**图例：**深蓝色是GPU相对于CPU的性能，红色是TPU相对于CPU的性能，橙色是TPU相对于GPU的性能，绿色是TPU相对于CPU的性能，紫色是TPU相对于GPU的性能.

1. **实验结论**

对于总体性能/瓦特来说，K80服务器是Haswell的1.2-2.1倍。对于增量性能/瓦特来说，当Haswell服务器电源被省略时，K80服务器的性价比是其1.7-2.9倍。TPU服务器的总性能/功耗比Haswell高出17-34倍，这使得TPU服务器的性能/功耗达到了K80服务器的14-16倍。

1. **结果解释**

从性价比角度和对神经网络负载而言，GPU的性能要远远超过CPU，TPU的表现要远远超过GPU。

1. **论文中的Figure 10**



1. **本实验的目标**

比较CPU、GPU和TPU三者之间的能耗比例。

1. **本实验的设计思想**

随着提供的工作负载利用率在0％到100％之间变化，论文测量了包括CPU，TPU和GPU在内的服务器的性能和功耗，并且以10％的工作量增量的桶形式收集。

1. **本实验的具体配置**

本实验所用的服务器和工作负载同上。

1. **本实验图中各种标记的含义**

**横坐标**是CPU、GPU或者TPU的利用率从0到100%变化。

**纵坐标**是每个die的功率（Watt/Die）

**图表类型**是一个折线图图.

**图例：**深蓝色是Haswell服务器的每个die的功率随利用率的变化；绿色是K80服务器的每个die的功率随利用率的变化；紫色是TPU的每个die的功率随利用率的变化。

1. **实验结论**

TPU具有最低的功率（每芯片40W），但能量比例较差：在10％的负载下，TPU的使用率是100％利用率时的88％。（这个简短的设计方案阻止了很多节能功能的出现。）毫不奇怪，Haswell在能源比例上是最好的：它在10％的负载下的功率是在100％利用率时的56％。 K80比TPU更接近CPU，在10％工作负载下的功率是满载功率的66％。不受计算限制的LSTM1执行类似的操作：在10％的负载下，CPU使用全功率的47％，GPU使用78％，而TPU使用94％。

1. **结果解释**

在这篇paper中的TPU方案中，TPU的能耗比例的表现是最差的，GPU次之，属CPU在能耗比例方面的表现最好。

# 结合课程的阅读论文的收获

首先，我学到了Roofline这个可视性能模型。这个在教材上有，但是老师好像没讲过或者老师讲了我们听到，反正我是人生中第一次看到这个东西的，就是在这个论文中看到的。Roofline模型是一种直观的可视化方法可以对比各种SIMD体系机构变体的潜在浮点性能。它将浮点性能、存储器性能和运算密度汇总在一个二维图形中。运算密度等于浮点运算数与所访问的存储器字节的比值。其计算方法为：获取一个程序的总浮点运算数，然后再除以在程序执行期间向主存储器传送的总数据字节。



图 ‑1 TPU外观图

 其次是了解了一种比GPU还要牛气的处理器，我都感觉快跟不上时代的变化了，从图四-1，我们可以看出，其中间突出一块很大的金属片，这便是为了可以很好地对TPU高速运算是产生大量的热进行耗散。

TPU的高性能还来源于对于低运算精度的容忍，也就是说每一步操作TPU将会需要更少的晶体管。在晶体管总容量不变的情况下，我们就可以单位时间在这些晶体管上运行更多的操作，这样我们就可以以更快的速度通过使用更加复杂与强大的机器学习算法得到更加智能的结果。我们在TPU的板子上看到了插条，所以目前Google使用TPU的方式是将载有TPU的板子插在数据中心机柜的硬盘驱动器插槽里来使用。

而且我觉得TPU的高性能还来源于它数据的本地化。对于GPU，从存储器中取指令与数据将耗费大量的时间，但是机器学习大部分时间并不需要从全局缓存中取数据，所以在结构上设计的更加本地化也加速了TPU的运行速度。

 最后，to my surprise，在系统机构的课上接触到了深度神经网络。TPU是和Tensorflow密切结合的一款专用处理器。而Tensorflow也可谓是红得发紫。Tensorflow其命名来源于本身的原理，Tensor（张量）意味着N维数组，Flow（流）意味着基于数据流图的计算。Tensorflow运行过程就是张量从图的一端流动到另一端的计算过程。张量从图中流过的直观图像是这个工具取名为“TensorFlow”的原因。Tensorflow有如下特点①高度的灵活性： TensorFlow不是一个严格的“神经网络”库。只要你可以将你的计算表示为一个数据流图，你就可以使用TensorFlow。②可移植性（Portability）：Tensorflow可以运行在台式机、服务器、手机移动等等设备上。而且它可以充分使用计算资源，在多CPU和多GPU上运行。③多语言支持：Tensorflow提供了一套易用的Python使用接口来构建和执行graphs，也同样提供了一套易于C++使用的接口（目前训练神经网络只支持python，C++接口只能使用已经训练好的模型）。未来还会支持Go、Java、Lua、Javascript、R等等。④性能最优化：TensorFlow给予了线程、队列、异步操作等最佳的支持，TensorFlow可以把你手边硬件的计算潜能全部发挥出来，它可以充分利用多CPU和多GPU。