**在网计算应用工作报告**

1. **简介**

SDN解决方案将控制平面与转发平面分离，并为我们提供了控制平面的可编程能力，目前人们提及最多的OpenFlow协议在逐渐的完善演化过程中，表字段和表类型不断的增加。白牌交换机在支持OpenFlow协议的版本更新上，面临着和传统交换设备厂商同样的困境——OpenFlow并不支持弹性地增加匹配域支持，协议新特性的支持所需要的成本大、时间周期长。同时，随着网络中新的协议不断出现，OpenFlow协议也必将变得越来越臃肿，表的扩展也必将变得越来越困难。转发平面编程语言P4和伴随它出现的新一代可编程交换机为运营商提供了转发平面的可编程能力，打破了硬件设备对数据转发平面的限制,让数据包的解析和转发流程也能通过编程控制，使得网络及设备自上而下地、真正地向用户开放。新一代可编程交换机的出现，它的灵活性和强大的能力为在网计算技术注入了新的血液，带来了很多新的思路。

网络拥有了计算能力，可以将分布式服务嵌入可编程交换机，这样可以充分的开发网络的潜力。原本需要在远端服务器中去争抢资源的分布式服务，可以在近端的交换机上直接解决。这提高了网络中的吞吐量，降低了包交换的延时。由控制器掌控全局，实现全局的管理功能，容错以及更新。再由P4编程语言自定义的灵活的包转发策略并且可以实现数据的key-value缓存。

发掘新的在网计算技术的应用也具有深刻的研究意义，希望通过本文研究进一步挖掘在编程交换支持下的在网计算技术的潜力。把在网计算技术应用到现在与人们生活息息相关的大规模数据的机器学习方向。把可编程交换机提供的缓存功能加入分布式机器学习架构，成为其中的重要一环。

现在已经确定将在网计算用于机器学习具体问题是利用因子分解机（Factorization Machine，FM）来解决广告预估模型。广告预估问题有数据规模大，并且数据稀疏的问题。在这类问题中，通常使用FM的方法来建模，然后使用分布式学习训练数据。在分布式机器学习中需要把节点分为worker nodes和sever nodes两类分别负责计算和存储全局数据。在每次迭代过程中，worker将自己通过随机梯度下降法（SGD，Stochastic Gradient Descent）的到梯度作为增量去更新server中存储的模型参数，而每次worker的迭代都需要从sever取得上一次更新的模型参数用于更新梯度。可以在这类问题中利用新一代可编程交换机的存储能力缓存每次更新的模型参数，以此将在网计算技术应用扩展到机器学习的领域中。

现阶段，正在将此问题通过mininet与P4语言进行软件仿真来验证其可行性与优势。在仿真中，我们使用p4c作为p4编译器，BMv2作为P4编译配置的软件交换机，PI（P4runtime的实现）实现控制平面对数据平面的控制。

1. **原理**

我们希望新一代可编程交换机能在其它的应用上也能大放异彩，首先我们想到的是它在机器学习方面的应用。

1. **分布式机器学习及参数服务器**

现实中，机器学习训练数据的数量可能达到1TB到1PB之间，而训练过程中的参数可能会达到10的9次方到10的12次方。因此几乎不可能在单台设备中完成对这类大规模机器学习数据的训练。所以我们需要使用多台服务器训练存储参数，这就是分布式机器学习的由来。

李沐提出了使用参数服务器框架的分布式机器学习方法。如图1所示，集群中的节点可以分为计算节点和参数服务节点两种。其中，计算节点负责对分配到自己本地的训练数据（块）计算学习，并更新对应的参数；参数服务节点采用分布式存储的方式，各自存储全局参数的一部分，并作为服务方接受计算节点的参数查询和更新请求。



图1.参数服务器架构

* Worker节点： 每个worker都存储一部分训练数据，并计算本地参数（一般为全局参数的增量）。并push到参数服务器，在新一轮的迭代中从参数服务器中pull更新后的参数。Worker节点之间没有通信，只跟对应的sever通信；
* Task scheduler节点：任务调度节点，将任务分配给worker，并监控他们的进度。如果添加或者删除了worker，它将重新安排未完成的任务；
* Parameter sever：接受来自worker的参数，更新全局参数；
* Server manager：负责维护一些元数据的一致性，比如各个节点的状态，参数的分配情况等。

参数服务器采用key-value的存储方式，用链式复制方式来提高系统容错性（即使用多个参数服务器以链的形式存储相同的数据，来提高容错性，应付可能出现的服务器故障问题）。我们使用参数服务器训练数据，并不是等到所有worker都完成本次迭代再发往参数服务器的方法，这样会造成先完成数据worker需要等待很久。我们使用异步训练的方法，当其中1个worker完成了本次迭代，将数据传给sever并不立即开始新的任务，而是设置一个作为最大延时时间，只有大于之前任务都被完成了，才能开始一个新的任务。这样保证了等待时间不会太长，也保证了每次迭代训练的数据都是有效的。

1. **广告预估模型及因子分解机**

在机器学习中，预测是一项基本的任务，所谓的预测就是估计一个函数

 （1）

该函数将一个维的实值特征向量映射到一个目标域中，例如回归问题中：，二分类问题中： 或者。在有监督学习中通常还有一个带标签的训练数据集

 （2）

其中表示输入数据，对应样本的特征向量，对应标签，为样本的数目。

在现实中很多应用问题会应用到预测模型，因为我们关注大规模机器学习，所以选定其训练数据规模较大的广告预估模型，通过个人特征以及浏览历史预估是否会点击当前广告（预测一个人对什么广告感兴趣，做广告推送）。但这类通常会有太过稀疏的问题。我们通过FM来建模。

回顾线性回归模型： 

 （3）

在此模型中，各个特征时独立考虑的并没有考虑特征之间的相互关系，如女生喜欢看化妆品类，而男生喜欢看球类。可以将上述模型推广到二阶：

 （4）

这样将任意两个（互异）特征分量的关系也考虑进来了。对上式来说，只有和都非0式，组合特征才有意义，并且从上面来看组合特征有个，如果特征很多组合特征的量也会特别多，并且非零的项会非常少，导致训练样本的不足，很容易导致不准确，为了克服这个缺陷，改变的表现形式：引入辅助向量

 （5）

并将改写为

 （6）

这样公式（4）就变成了

 （7）

上式还可以改写为：

 （8）

上面实质上的变换，对应与一种矩阵分解，于是最终得到的模型方法就叫做Factorization Machines方法。通过FM不仅解决了数据稀疏的带来学习问题，还降低了模型的复杂度。

1. **随机梯度下降法**

通过上面的预估模型，可以得到相应的损失函数。我们使用随机梯度下降法来使得损失函数最小，得到需要估计的参数。

梯度是指函数增加最快的方向，所以求损失函数的最小值，只需要将变量（模型参数）沿着梯度的反方向，一次次迭代，便可以得到损失函数最小值，损失函数最小值时的模型参数便是训练出的最优模型参数。

但是梯度下降法每次更新都需要遍历所有的data，当数据量太大或者以此无法获取全部数据时，这种方法并不可行。解决这个问题的基本思路是：只通过一个随机选取的数据来获取梯度，以此对模型参数进行更新。这种优化方法叫做随机梯度下降法。随机梯度下降法对比梯度下降法来说虽然单次迭代的梯度方向可能会有很大偏差，但足够多次迭代方向的期望与一次遍历所有数据得到的方向是一致的。SGD相对来说要快很多，但是也存在问题，由于单个样本的训练可能会带来很多的噪声，使得SGD并不是每次迭代都向着整体最优方向。因此在刚开始训练时可能收敛的很快，但是训练一段时间后就会变得很慢。再次基础上又提出了小批量随机梯度下降法（我们说的随机梯度下降其实指的是小批量随机梯度下降法），它是每次从样本中随机抽取一小批进行训练，而不是全部数据。这也契合参数服务器框架训练参数的思想。

1. **交换机缓存的参数服务器模型**

结合之前的参数服务器与当前在网计算技术的发展，我们希望能通过交换机在参数服务器的框架中扮演缓存服务器的角色，将一部分worker pull参数从应用层转到第二层的交换机，以此提高大规模机器学习的效率。并且减轻参数服务器的带宽压力。起到负载均衡的作用。新的框架如图2所示。



图2.交换机缓存的参数服务器模型

在以往的参数服务器框架中新加了switch与controller，switch是用于缓存每一轮新的参数，switch以key-value方式存储latest值，初始值为0，Latest为1表示当前switch的值是最新的，0表示当前switch缓存的参数非最新值。controller用于switch的故障恢复。

新的参数服务器工作逻辑（以随机梯度下降法举例）：

1. 首先Task Scheduler将training data分配给多个worker，worker计算出第一轮梯度，将新得到的梯度push给参数服务器。之后等待之前开始工作的worker工作完成。才可以进行新一轮迭代。
2. 参数服务器得到worker发来的梯度更新模型参数，并存储下来。
3. 当第一个worker可以进行新一轮迭代了，便去pull模型参数。第一个worker经过switch发现switch中latest的值为0，便向参数服务器去pull参数，并在返回时将latest设1，并在switch出缓存一部分参数。其余worker需要pull参数发现latest为1，便可以在switch中pull一部分参数。
4. 当参数服务器发现参数被更新便发送指令把switch中的latest的值设为0。直到新的worker需要pull参数。循环这个过程直至训练结束。

上述过程保证，每一轮最新的参数值只有第一个需要的worker需要从参数服务器pull全部参数。

1. **软件仿真**

我们现在的工作是通过mininet软件在虚拟机上做一个demo验证我们的想法。软件环境抽象如图3。



图3.仿真环境结构

如图，我们写好xxx.p4代码，通过 p4c 这个 p4 compiler 将p4代码编译成为p4交换机可以理解的各种”机器代码”。如果目标交换机是 bmv2 , 那么p4c将生成 .json文件

* p4c是一款 p4编译器；
* BMv2是支持P4编程的软件交换机；
* PI是P4 runtime的实现，用于Control Plane对数据平面的控制。
* mininet的功能是构建一个虚拟的网络拓扑。 它通过linux内核的一些特性(net命名空间)，在一个主机上划分出多个虚拟网络空间，各个网络空间之间相互隔离，有自己的端口, ip等等。mininet让一个或者多个vhost（虚拟主机）, 软件交换机(如ovs, bmv2)等。以进程的状态分别绑定在这些网络空间之中，共同构成一个进程级别的虚拟网络拓扑。需要注意的是这些进程级别的主机和交换机他们只是网络上的隔离，而文件系统则是共享主机的文件系统。