FlexPS: Flexible Parallelism Control in Parameter Server Architecture

FlexPS：在参数服务器框架下的灵活的并行控制

摘要：在现实中，一些系统（比如PS和Petuum等）都是在PS框架下实现的，但是这些系统都不支持在运行时改变并行度，根据动态的工作量来有效的执行机器学校任务。所以作者希望设计一个新的系统，叫FlexPS，它引入了一种新的多阶段抽象来支持灵活的并行控制。提出了阶段调度器、直接模型传输等优化方法。作为一个一般和完整的PS系统，FlexPS集成了许多优化，这些优化不仅限于多级机器学习。

1. 介绍：现在所有的PS系统都忽略了一个重要的特性，灵活的并行控制，这有关于动态工作负载。现在的PS系统中只采用一个恒定的并行度（用于任何任务），不支持在运行的时候更改并行度。缺乏灵活的并行度会极大地阻碍分布式机器学习地性能，会导致长时间地运行，会浪费资源。

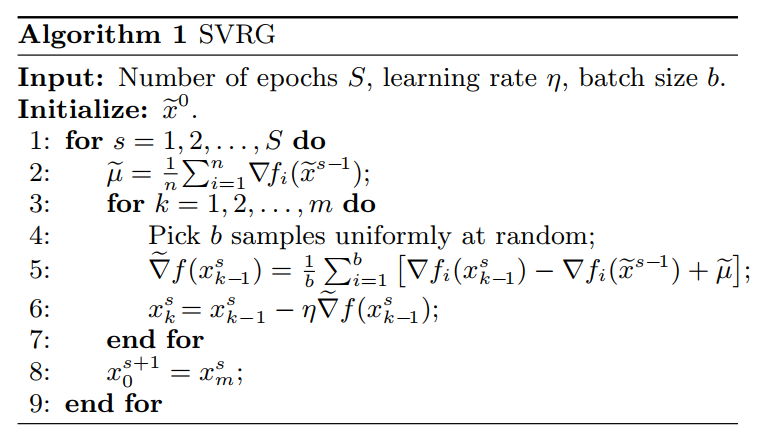
作者提出了一种新的迭代式机器学习的多阶段抽象，基于这个方法设计了一个新的PS系统，FlexPS。在多阶段抽象中，一个机器学习任务被看作由一系列阶段组成的，阶段可以有明显的并行度。一个动态工作量的机器学习任务能映射成多阶段，每一个阶段根据它的工作量来有自己的并行度。作者在FlexPS中提供了一个模块，可以在运行时自动调整并行度。（详细）

FlexPS是一个普遍的和完整的PS系统。这篇文章的主要贡献：

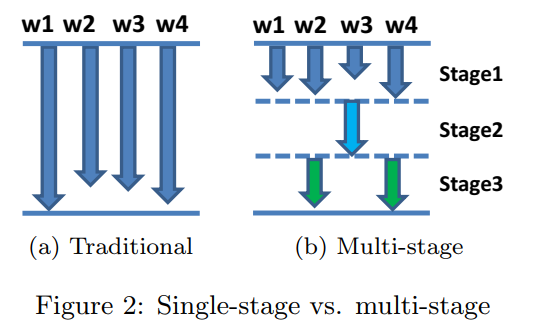
1. 确定了动态工作负载的灵活并行控制算法的需要
2. 设计一个新颖的多阶段抽象。
3. 一个新的PS系统和定制的系统设计
4. 进行广泛的实验来评估FlexPS的性能
5. 灵活并行控制

在PS系统中的一个基本权衡是计算时间和通信时间。每一次迭代的延时是指计算时间和通信时间的总和。现存的PS系统中都是一个固定的并行度，而且不支持在运行时改变并行度。

提出了一个例子：SVRG。



1. 多阶段抽象



单阶段和多阶段的区别如上图所示。

阶段是调度的基本单元，用户可以为每个阶段指定资源分配（包括并行度）。

用使用SVRG算法的logisitic回归作为例子，SVRG是典型的动态工作量算法，而且它的全梯度步骤比随机步骤有更高的工作量。

过程：第一、将任务映射成两个阶段，全梯度阶段和随机阶段。全梯度阶段通过扫描整个训练数据集来计算整个梯度，并且可以分配大量的工作节点来处理繁重的工作负荷。

1. 随机阶段利用小批量的数据样本来对模型参数进行更新。因此只需要少量的工作节点。

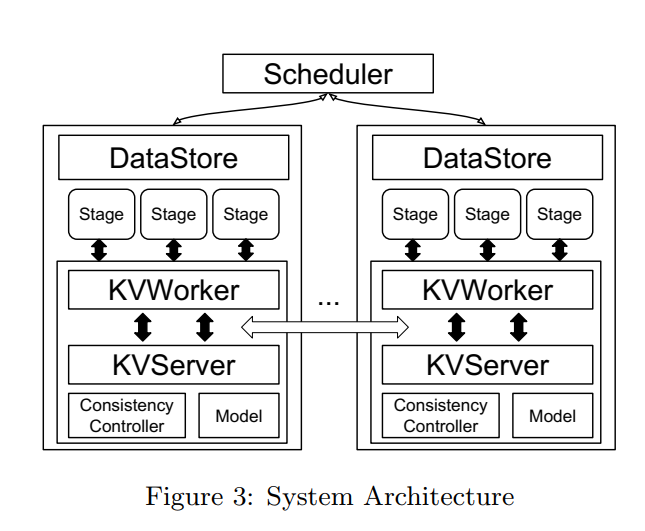
两点好处：1.使我们能够更好地利用数据局部性。如果不同的阶段操作数据集的不同部分，则可以将它们分配到所需数据驻留的机器上，以避免在网络上传输训练数据。2.简化了多任务的细粒度调度，因为任务的一个阶段比任务轻得多，不同机器学习任务的各个阶段可以在集群中同时运行。

困难：难以实现或者是没有效率。解决方法：1.将每个阶段映射到一个任务，但是任务之间的转换是困难的2.动态地杀死工作线程等，但是，这需要修改许多系统组件，并不是很有效。

问题：1.怎样用有效地方式来安排阶段。2.如何确定各阶段之间地有效过度3.怎样减少通信量。

1. 系统构架和优化

系统构架图：



调度程序位于master的核心，通过考虑阶段规范、从属可用性、并行度、数据局部性等各种因素，负责将任务阶段分配给从服务器。为了有效的利用集群中的资源，调度器支持多个任务同时运行。当新的阶段被分配时，会根据给出的规范产生新的线程来运行。会通知master当一个阶段结束的时候。从属节点仍然当作PS。每一个从属节点中都有KV存储模块，包括KV工作线程和KV服务器线程。用户定义的函数链接到KV-worker线程来发出非阻塞get()和put()。

多阶段的优化：

在Flex PS中的基本调度单元是阶段。

Datastore是中存储的是二维静态向量。每个本地worker只对特定的行进行写入，每个worker都可以访问整个本地数据存储来进行训练。导入任务正常情况下使用大并行度来加速导入进程。