**4 多维度测试**

**4.1** 多维度测试概述

分布式机器学习系统的架构从上到下可以分为实际应用，机器学习算法，计算框架和物理集群，从计算框架层面和物理集群层面上看，影响系统扩展性的因素主要包括计算框架的数据并行度和任务并行度，物理集群包括处理器CPU的个数和内存大小以及磁盘I/O.

|  |  |
| --- | --- |
| 影响物理集群和框架的扩展性因素 | |
| 计算框架 | 数据并行度：输入数据的分片个数，在shuffle计算中，该参数决定了map端的数据分片 |
| 任务并行度：在shuffle计算中，该参数决定了reduce端的数据分片 |
| 物理集群 | core:物理CPU个数 |
| 内存大小 |
| I/O（磁盘I/O） |

**表1** 影响扩展性的因素

这些因素决定了在计算框架和物理集群层面的分布式机器学习算法的扩展性。当前工作主要评测任务总的执行时间，进一步分析发现，任务的执行时间可以分为任务处理时间和系统开销时间.

|  |  |
| --- | --- |
| 任务执行时间 | |
| 数据处理时间 | 数据预处理时间 |
| 模型训练时间 |
| 预测时间 |
| 系统开销时间 | 任务调度时间 |
| GC时间 |

**表2** 执行时间

本文针对以上当前测试工作的不足，在测试方法上充分考虑数据并行度、任务并行度和物理集群core（CPU）数量以及内存大小，采用细粒度的时间维度，搜集包括数据处理时间和系统开销时间，提出了一种多维度、细粒度的测试方法。为了充分考虑数据并行度，任务并行度和物理集群core数量以及内存，同时兼顾细粒度的时间分析，本文需要应对以下挑战，第一是多维度组合测试空间爆炸，在测试方法上，数据并行度、任务并行度和物理集群core以及内存大小的不同参数会组合成爆炸式测试情境；第二是细粒度的时间分析，当前框架的监控系统（UI或者web监控），往往提供的是端到端的执行时间，即从任务提交开始到任务完成所需要的总时间，没有提供较为详细的调度时间，等待时间等时间粒度；影响系统扩展性的因素有很多，包括CPU，任务并行度，数据并行度等，如何从这些因素中确定影响系统扩展性的瓶颈也是一个挑战。

**4.2** 多维度测试方法

**4.2.1** 面向集群和计算框架的横向测试

在给定数据量的时，物理集群资源可变参数是物理集群core数量和内存大小，计算框架配置可变参数是数据并行度（输入数据分片个数或者shuffle map数据分片个数）和任务并行度（shuffle reduce数据分片个数），组合测试空间爆炸，需要进行测试空间的削减。基于对计算框架的数据流和物理执行计划观察发现，增大数据并行度，输入分片的数量增大，shuffle map每个分片的数据成倍的减小，在确定物理资源core时，每个core处理的数据是不变的。但是在shuffle reduce数据由于shuffle算法和reduce数据分片个数，其数据分布是不同的，因此每个core处理的数据是不同的。在不考虑网络延迟、任务调度的情况下，仅改变数据并行度，对系统的执行时间是没有影响的。基于以上发现，可以对测试空间进行削减，把测试空间从原来的物理集群core个数、数据并行度和任务并行度削减为任务并行度和物理集群core与执行时间的关系。面向集群和计算框架的横向测试物理集群core和任务并行度与执行时间和系统开销之间的关系。

**4.2.2** 面向系统开销的混合测试测试

横向测试评价时间与物理资源的变化，在此基础上本文提出混合测试的方法，用来衡量overhead和系统扩展之间的关系，即：同时按比例增加系统资源和数据量，测试系统的overhead。保证任务并行度不变，同时按比例增加数据和物理core数量，理想情况下（没有任何开销的情况下），执行时间是不变化的，但是实际测试结果中，由于存在网络，调度等因素，执行时会发生变阿虎，那么两者之间的差值就是系统的overhead。

**4.2.2** 面向瓶颈分析的纵向测试

通过横向测试，如果算法具有良好的线性扩展性，那么结合上述的实验算法没有线性扩展性，在平静点再进行纵向测试。

固定任务并行度和core数量，增大或者减小数据量，观察算法的执行时间，分析影响系统扩展性的因素；固定数据量和任务并行度，逐步扩充物理资源，观察算法的执行时间，分析影响系统扩展性的因素，同时结合机器学习算法的迭代特性和特定算法自身的特性，分析分布式机器学习算法扩展性的瓶颈原因。