Key notes.

包总 计算所：

受本人研究方向不同的原因简要记录：

关于敏捷开发之类的东东以前听了一些报告，我勉强算是比较熟悉了

下一步工作，开源EDA软件更精细工艺的支持（对于28nm）还没有较好的支持能力。

验证工作：可不可以利用社区的力量来做好验证，减少因为敏捷芯片开发设计造成的验证困扰和问题。

刘譞哲 北京大学：

大模型系统软件栈

-
- 大规模：巨大成本和巨大投资
- 大范围：算力需求增长，东数西算

传统算力发展速度远远低于计算需求，大模型时代的算力危机。

一张单卡很难完成训练需求，必须依赖分布式，为此经济上的开销或将成为天文数字。

系统软件在这边就起了承上启下的中坚作用。在上需要通过应用软件来提供高效友好的编程与人机接口，在下则需要尽可能发挥硬件能力以使得其逼近物理极限。

在系统软件上层来说会有几个必要性东西：

- 支撑框架与工具链（这个环节也因为AI的发展出现了颠覆，会增加AI模型训练等新环节）
- 资源分配与调度管理
- 编译优化，算子库与运行时
- 异构硬件虚拟化和加速

如何支持异构尤其是国产硬件？

- 国内软件工具生态落后了很多的大版本
- 国产硬件发展迅速，但是对应AI所做的系统软件则心有余而力不足

如何提高资源利用率？

- 阿里云：90%的GPU/CPU利用率低于50%，90%的内存和显存利用率低于30%

数据处理->模型训练->模型调优->模型推理

如何高质量数据处理？

- 切分数据和冗余消除，有效查询和高效编排

如何并行处理支持大规模训练任务？

- 任务模式选择与资源共享
- 容错率与通信效率优化

如何做到模型的优调？

- gpt的优调需要大量的成本，但是大模型的调优会有很明显的优化，比如LLaMA的表现

如何准确高效地模型推理？

- 推理时怎样减少时间
- 可不可以用cache再用向量化查找来加速（上交李明煜在HeDB的未来工作中提到）
- 等等

如何在轻量设备中使用？

- Edge Server怎样更好使用大模型？
- 我有个问题，这边会不会引入一些安全问题？

如何控制异质开发环境的复杂性？

- 模型相关处理会引入新的工具链

如何跨域算力调度？

- 统一的资源调度安排这可不好做，关键词：sky computing

更好的用户体验？

租卡5小时，使用2小时？是不是会浪费算力，算力是很奢侈的。（似乎没有一个系统这么做，买断式服务真的很奢侈）

近期的工作

scalable adaptive affordable system，两个原则：

- 泛在资源互操作的服务化，横向做链接
- 泛在原生的开发运维一体化，纵向提供给开发者更好的使用体验，可不可以直接提供一些RPC之类的API把事情做好？

ElasticFlow，可以简单关注一下

问题：

- 自适应资源调整问题怎么处理，怎么根据任务来看资源使用可能性？到底是用CPU多还是用GPU多？怎么去看？
- 监控当前资源，分析程序代码，可能需要依赖部分编译技术，得到部分先验知识，北大有一个工作和这个相关。

Session1： 机器学习系统

陈文艳 深先院：

GPU调度工作相关：

当前优化方式： time 和 spatial的Multiplexing。

当前一些解决方法上的问题：

- AntMan - 过于backward，有些场景没法用
- MPS and MIG - 软硬件隔离，前者不够健壮，死一个软件可能会崩，而后面的MIG则只比较支持高端GPU，分配任务时，方式也比较固定，导致使用起来不够灵活，场景也不够多元

折中：IADeep

比较动态，粒度比较小，能够相对灵活地分配GPU资源

方法

找到任务训练时的资源开销奇点

arch：

- Online Scheduler：根据Task stats来做预测和在线资源调度
- Tune: Tune configurations，减少干扰

量化：

建模了一个用于衡量性能的参考标准

形式化方法的思想可以

问题：

隔离与不隔离的方式也似乎在里头提到，似乎是通过time validation来做区分。

李明真 计算所：

GPU资源的可扩展与异构性，这个是一个Gap。

问题：

- 很长的任务排队时间，需要等待GPU板卡准备完毕
- 容易被干扰，为了提高集群资源利用率，一个任务可能会借用其他的算力。可能会触发对于其余人资源的抢占。

实验观测： 模型精度上存在一些不一致，会在部分微妙场合，如自动驾驶，造成比较大的隐患。

利用线程抽象来做，尝试把不一致性给消除掉，只能做到常见问题的缓解。

刘方鑫 上海交大：

量化神经网络压缩

基于观察：

矩阵稀疏化处理，量化后的参数80%其实还是在INT4的表示范围之内。

引入SPARK一个新的编码设计，实现自适应的存储通信开销的降低。

特殊设计的编码器

有超过95%以上的数据都是可以做到Spark编码后无损的

管乐 上海交大：

神经网络剪枝：动静态剪枝需要在不同网络上独立实现算法，比较耗费时间。

然而，剪枝的语义完整，可以直接通过`API`，完成对应的接口实现，来尝试做。不过这边的工作量依旧很大。

传统技术中有一个插桩，以简化应用开发的流程。

三种基本语义似乎都是可以通过插桩来尝试做的，那么在DNN层面，是否能够提供一个插桩机制来减轻开发的负担？

插桩当前所需要的信息：

- 注册
- 分析流程
- 插桩流程

可以简化开发流程

问题：

支持分布式的插桩吗？

- 现在的evaluation还没有做分布式，开销比较大

对于特别动态的语言怎样减少开销？

- runtime动态来支持pytorch上的插桩工作，来一个op请求，就插上一个桩，以减少开销。不过性能上确实还是一个比较大的问题。

Session2: Model Inference

郭力维 华为：

在AI环境下的系统软件下是否有新的生态？考虑一下设备侧的事情。

设备内的NLP推理有更高的隐私性

需要什么：

- 比较大的模型：对于端侧来说，这个空间消耗已经很大了
- 低延时：还是希望用户使用起来比较快

稻草人：

- hold in memory：洛阳纸贵
- load before exec: Long delays of modle loading，相比较而言推理的时间非常短

要不要做流水线？

- 有很多时间都在空转，对于transformer大模型上（计算密集型），开销本身就在IO上放了很多了，不可接受

弹性流水线机制：

- 把流水线的bubble给挤出来：
- - model sharding: 小碎块
- - Elastic Pipleing: 弹性流水线，减少Stall

Model Sharding

选用transformer中的压缩过的几部分layer，而不是全部。

通过采样切分，构建NMK这样一个三维立体混元劲

IO pipeline planning： 把IO资源优先分给最重要的shard，打分是通过分配bit位隐性实现的

端到端的结果： 不成比例的IO与计算比重，有极大部分的时间浪费掉了，IO一直保持饥饿。

而Elastic Pipiling的效果变得比较好。

问题：

是否适用于很多NLP模型？对于模型是否有要求？

- 模型必须要支持切块，因此需要一个数据特性，华为诺亚实验室的某个模型似乎是有这个特性

在设备侧做切割和决策的开销大概是多少？

- load开销因为做了切割，而decompress则是通过并行化处理。

一层流水线，带宽，在设备侧中也会有影响，怎么解决？

- 针对bert模型，内存带宽对于bert来说并不算是很大的瓶颈，暂时还没有考虑到这些

模型压缩的量化方法，data loss之类的问题？

- simulating，并不是真正把浮点数转化成整形，而是通过参数的统计分布来找到比较好的压缩。对于精度的影响并不大。

苗旭鹏 CMU

大模型的费用巨大，国内的云计算平台有便宜的实例模型，spot实例，即竞价实例。

但是他们容易被抢占，我们应该怎么做？

想把大模型场景移植到spot实例中存在难度。

- 动态重并行：结点数量变化太频繁了，以前工作确定并行策略，很慢
- 上下文的迁移：很慢
- 宽限使用时间：怕给中断然后给扬了

这篇文章感觉可以去看一下哈

把四个步骤模块化组成一个系统。

并行控制单元：

首先满足吞吐要求，再选延时最低，没有的话就找吞吐最大的。

设备映射单元：

各个参数应该放在哪一个物理设备上，希望最大化复用当前设备已经持有的上下文信息

带权二部图匹配建模

迁移规划单元：

使用贪心算法微调顺序，以防止超过显存限制。

中断筹划单元：

主动中断推理，并且添加中断恢复机制，以进行下一个迭代计算