**人工智能算力评测程序设计方案**

**清华大学**

**2022年12月5日**

## 一、研究背景

随着近年来人工智能应用的大发展和在大规模算力上的普及，公众需要一个简单有效的指标来回答哪套系统的人工智能算力更强以及整个领域的发展状况如何。一个好的指标正可以引领领域的健康发展。然而传统的高性能计算机评测方法和体系与当前人工智能需求的性能并不完全一致。举例而言，Linpack是一个目前被广泛采用的高性能计算机双精度浮点运算性能基准评测程序，国际超算TOP500榜单就是Linpack值来排名的。然而典型的人工智能并不需要双精度浮点数运算，例如人工智能训练以单精度浮点数或16位浮点数为主，推理以Int8为主。

对大规模人工智能算力来说制定一个简单有效指标和测试方法并不容易。首先单个人工智能训练任务很可能达不到全机规模。即使能达到，可能在训练时间和准确率上没有改进，因此使用全机规模是没有意义的。再则如果要测试规模变化巨大的人工智能集群计算机，测试程序必须是规模可变的。而什么样的主流人工智能应用是规模可以大幅变化的，是一个必须要明确的问题。最后如何考虑准确率是相对于传统高性能计算基准评测的一个最大差别。是否需要使残差小于给定标准，还是将准确度计入分数统计，都需要明确。

目前，在人工智能性能基准测试这个领域已经有很多探索，比如谷歌等公司主导的MLPerf，小米的MobileAI bench，百度的Deepbench，中国人工智能产业发展联盟的AIIA DNN Benchmark，在双精度的Linpack基础上改成混合精度的HPL-AI等，都不能很好的解决这些问题。例如下图所示，根据MLPerf公开发表的数据可知，MLPerf程序在百卡以上规模测试下扩展性就会出现下滑，在千卡级别到达评测体系的扩展性瓶颈，即该评测程序已经不能区分不同系统在该规模下人工智能算力的差异。



MLPerf 可扩展性

为了解决以上问题，我们设计了AIPerf500与AIPerf-MoE大规模人工智能基准测试程序。AIPerf500采用大规模自动化机器学习（AutoML）作为基础框架，实现：1. 只给出一个分数（类似Linpack），分数近似具有线性性；2. 可变问题规模，可以适应从单机单卡内存规模的变化到多机多卡并行策略的调整；3. 计算有人工智能意义，通过网络结构搜索找到更好的人工智能模型。具体而言，AutoML是通过算法自动搜索合适的神经网络模型结构，找到针对特定任务效果最好的解。故而所需的计算资源极高，基础算法也包含训练模型本身的过程，而且任务自身是深度学习任务设计中的关键环节，能够直接体现超算系统在深度学习方面的综合性能（包括训练、推断）。AutoML具有足够的并行度，常常需要同时训练大量候选模型来对结构进行评估。搜索的结果虽然有一定的随机性，但整体上能找到的解的优劣程度随着搜索所消耗的计算量而逐渐改善。AIPerf500的设计达到了以下四个关键的设计目标：

**1、一个统一分数**

AIPerf500能够报告一个分数作为被评测计算集群系统的评价指标。AIPerf500目前的评价指标是Tops，即平均每秒处理的混合精度AI浮点操作数。使用一个而不是多个分数能方便进行不同机器的横向比较，以及方便公众的宣传。

**2、可变的问题规模**

人工智能计算集群往往有着不同的系统规模，差异性体现在结点数量，加速器数量，加速器类型，内存大小等指标上。因此，为了适应各种规模的高性能计算集群，AIPerf500能够使用AutoML进行问题规模的变化来适应集群规模的变化，从而充分利用人工智能计算集群的计算资源来体现其算力。

**3、具有实际的人工智能意义**

具有人工智能意义的计算，例如神经网络运算，是人工智能基准测试程序相较于传统高性能计算机基准测试程序的重要区别，也是其能够检测集群人工智能算力的核心所在。目前，AIPerf500在ImageNet数据集上训练神经网络来运行计算机视觉应用程序。

**4、包含必要的多机通信**

网络通信是人工智能计算集群设计主要指标之一，也是其庞大计算能力的重要组成部分。作为面向高性能计算集群的人工智能基准测试程序，AIPerf500包括必要的多机通信（如任务的分发、结果的收集与多机训练），从而将网络通信性能作为最终性能的影响因素之一。

AIPerf-MoE 基准测试程序基于 FastMoE 开源框架, 以 MoE-GPT 模型训练为负载, MoE 是一种新兴的模型结构, 是当今将预训练模型规模扩展到万亿级参数以上的唯一方式。 相比于传统的稠密模型, MoE 模型通过稀疏激活专家的方式, 在不提升计算量的同时扩大参数规模, 从而使得大模型的训练与推理成为了可能。 MoE 模型在语言预训练模式上对于模型质量有较大的提升。 此处选用的 GPT 模型是一种生成语言模型, 可以根据上文自动生成下文。

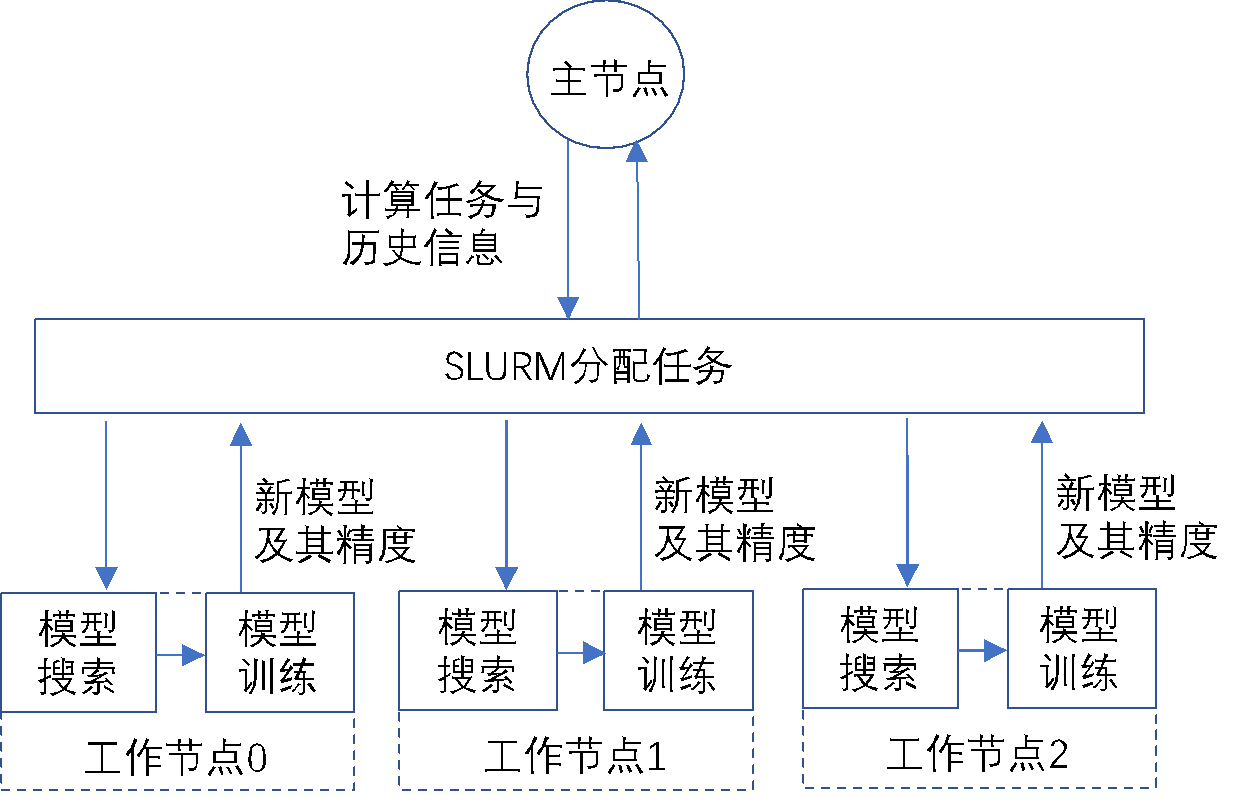
## 二、框架设计与关键技术点

AIPerf500基于微软NNI开源框架实现，以自动化机器学习（AutoML）为负载，使用Network Morphism网络结构搜索和TPE（Tree-structured Parzen Estimator）超参搜索来寻找精度更高的神经网络结构和（或）超参数。

用户可以通过配置文件指定AutoML的相关参数，如训练使用的batch size、最大epoch数、学习率、最大搜索模型总个数、最长搜索总时间、最大同时搜索模型个数（并发数）等多个参数。

通过配置文件指定AutoML任务后，用户可以在主节点上启动AIPerf500。如下图所示，启动后，主节点将持续地将AutoML任务以及当前最新的历史信息通过SLURM资源管理系统分发给各个工作节点，其中，历史信息包括已经被工作节点完成的AutoML任务的网络模型及其精度。工作节点接收到任务与历史信息后，将根据历史信息开始搜索并生成一个新的网络模型。模型生成完成后，工作节点调用后端深度学习框架（默认使用Keras和Tensorflow）进行神经网络训练。训练过程中，使用NFS（Network File System，网络文件系统）共享数据集文件，方便所有工作节点并发读取。

工作节点根据一定策略（如训练ephch数达到配置文件指定的最大epoch数或连续10个epoch没有精度提升）停止模型训练，并将此网络模型及其精度发送给主节点，主节点更新历史信息，并继续发送新的任务和历史信息。当任务完成数达到配置文件中指定的最大任务数量或AIPerf500启动时间到达配置文件中指定的总时长时，AIPerf500结束。



AIPerf500执行流程

在AIPerf500执行过程中，所有工作节点上的任务都是异步执行的，即一个工作节点完成其任务时，不需要等待其他节点，可以直接将新网络模型及其精度发送回主节点，并从主节点接收下一个任务以及新的历史信息。

模型搜索与生成是AutoML中至关重要的一步，AIPerf500 目前使用Network Morphism网络结构搜索和TPE 超参搜索来寻找精度更高的神经网络结构和（或）超参数。Network Morphism使用了网络形态技术来搜索和评估新的网络结构。在AIPerf500中，工作节点首先会从父网络生成几个子网络，然后使用贝叶斯算法，从网络的历史训练结果来预测子网络的模型精度。接下来，选择预测模型精度最高的子网络进行训练。从父网络生成子网络包含一些层的变换，如变宽，变深，或在层中增加跳跃连接。TPE是一种基于序列模型优化的方法，它能够根据历史模型精度来按顺序构造模型，来估算超参的性能，随后基于此模型来选择新的超参数。

显然，历史数据的数据量将随着任务完成数增多而增多，且使用的工作节点越多，历史数据量的增长速度也就越快。由于模型搜索所需要的时间与历史数据量密切相关，当集群规模非常大的时候，AIPerf500执行后期可能需要花费非常多的时间用于模型搜索，这将导致得到的集群AI性能指标下降。AIPerf500采用了多种方法来减少模型搜索所用时间。

一种方法是当一个工作节点在进行模型训练的时候，其异构加速器被占用，而CPU基本是空闲的，此时CPU可以使用目前的历史数据提前进行模型搜索与生成，即使用模型训练的时间来掩盖模型搜索与生成的时间，不过这种方法的一个缺点是由于提前进行模型搜索，使用的历史数据并不是最新的，可能导致模型搜索效果较差；另一种方法是使用异构加速芯片，例如GPU，加速模型搜索与生成的过程。模型搜索与生成需要对旧模型与新模型进行编辑距离的比较从而找到预计最好的模型进行后续的实际训练。而传统编辑距离的计算由于依赖复杂难以并行，用异构加速芯片加速则是通过基于多边形变换的方法来消除复杂依赖，将编辑距离计算进行并行化，同时采用多边形拼接的方式来增大交给异构加速芯片的负载从而充分利用加速芯片的计算资源。这个方法是直接对模型搜索生成过程进行加速，因此保证使用的历史数据是最新的。

**三、面向预处理模型的可变规模评测程序**

AIPerf-MoE 基准测试程序基于 FastMoE 开源框架, 以 MoE-GPT 模型训练为负载。AIPerf-MoE 目标的 MoE 模型的层数, 专家数, 隐层大小等参数均以根据机器规模进行调整, 从而充分利用机器的内存空间来存储模型, 并适应于从 GB 到 PB 不同级别的内存空间的机器。 AIPerf-MoE 使用了模型并行和专家并行的两级并行, 将进程两级分组。 在模型并行组内, 进程之间进行 all-reduce 通信, 在专家并行组内, 进程之间进行 all-to-all 通信。 而通信的带宽对于最终得分影响较大。 运行测试时, 可以根据被测机器的网络拓扑结构, 对于并行方式和进程映射进行调整, 从而充分利用网络, 降低通信开销, 以获得更高的分数。

AIPerf-MoE 通过检查 loss 的方式对测试进行有效性验证。 具体来讲, AIPerf-MoE 在 enwiki8 小数据集上进行训练。 由于典型模型规模通常远大于数据集大小, 故模型会 “背” 下数据集, 并产生过拟合。 这样的现象可以说明训练系统是有效的。 相反, 若模型无法在该数据集上过拟合, 即无法达到较低的 loss 要求, 或 loss 无法下降, 则认为该训练系统的数值正确性存在问题, 并认为该测试无效。

混合精度是大规模 AI 模型训练系统的发展趋势之一。 AIPerf-MoE 支持混合精度训练, 并将 16, 32, 64 位的浮点操作均视为一次 Flop。 GPT 预训练模型对于低精度训练具有较好的适应性。 若混合精度训练方案可以通过上述的过拟合测试, 则认为该混合精度方案有效。

由于专家选择具有一定的自由性, 在训练 MoE 模型时在专家之间往往会发现较为严重的负载不均衡现象。 为了使被试系统专注于优化计算和通信性能, 而非优化模型本身, 在 AIPerf-MoE 中使用了 SWIPE 完全负载均衡策略。 具体来讲, SWIPE 策略首先使所有输入自由选择最适配的专家, 然后对于每个专家收到的输入进行统计。 当一个专家收到的输入超出限制时, 超出部分的输入将会被收集起来, 并重新分配给未达到限制的其它专家, 从而使得所有专家收到的输入完全一致。 实现上, SWIPE 算法可以通过一个低复杂度的高效轻量化算法实现, 从而最大限度地减小了负载均衡对基准测试评分的影响。

**四、基于大规模预训练模型评测程序**

随着BERT、GPT等预训练模型在自然语言处理、计算机视觉、多模态语言模型等多种场景取得优越的效果,大规模预训练模型成为了人工智能领域的一个重要研究方向。预训练通过自监督学习从大规模数据中获得与具体任务无关的预训练模型。为了提高预训练模型的泛化能力，近年来预训练模型的一个趋势是扩大预训练数据和模型参数。大规模预训练模型带来了新的挑战，目前已有的预训练模型参数已经到达万亿规模，从而对人工智能算力和集群性能有着更高的要求。对于自然语言处理任务，AIPerf选择经典的预训练模型BERT训练作为训练任务评估自然语言处理的性能。

**五、基于深度学习推荐模型评测程序**

基于深度学习的推荐模型是产业界十分重要的模型，在互联网的广告、推荐等场景具有十分广泛的应用。近年来，随着模型结构的发展和互联网数据量的增长，基于深度学习的推荐模型参数量越来越大，计算负载越来越复杂，训练实时性要求越来越高。因此，使用高性能的人工智能算力进行深度推荐模型的训练是业界非常重要的提升生产效能的手段。对于该种负载，AIPerf选择了广泛使用的的深度推荐模型DeepFM来评估人工智能训练系统处理深度推荐模型训练任务的能力。

## 六、测试内容及方法

AIPerf500已开源在https://github.com/thu-pacman/AIPerf/tree/Atlas。AIPerf500以Resnet50为初始网络进行Automl算法，不断生成新神经网络并使用Imagenet数据集进行训练。用户可以通过配置文件指定AutoML的相关参数，如训练使用的batch size、最大epoch数、学习率、最大搜索模型总个数、最长搜索总时间、最大同时搜索模型个数（并发数）等多个参数。

在环境方面，AIPerf500需要使用网络文件系统、Slurm作业管理系统和神经网络训练系统。对于常用的加速硬件（如Nvidia V100和A100），我们提供了docker镜像，便于用户搭建环境。

AIPerf-MoE 提供了基于 PyTorch 的针对 NVIDIA GPU 平台的标准实现, 并已开源在 https://github.com/thu-pacman/AIPerf/tree/Atlas/AIPerf-MoE-MindSpore-master。 该实现基于 NVIDIA 的 Megatron-LM 平台进行 GPT 模型的训练, 使用 FastMoE v0.3.0 来实现 MoE 模型, 并使用 NCCL 通信库进行通信。

具体搭建环境和测试步骤在开源代码中有详细说明。

(1)BERT预训练

BERT的主干结构为Transformer。对于BERT\_base，Transformer包含12个编码器模块，每个模块包含一个自注意模块，每个自注意模块包含一个注意模块。对于BERT\_NEZHA，Transformer包含24个编码器模块，每个模块包含一个自注意模块，每个自注意模块包含一个注意模块。BERT\_base和BERT\_NEZHA的区别在于，BERT\_base使用绝对位置编码生成位置嵌入向量，而BERT\_NEZHA使用相对位置编码。BERT使用WikiExtractor作为预训练语料。用户可以通过配置文件相关参数，如训练使用的batch size、最大epoch数、序列的长度、中间层数、注意头的数量等多个训练参数。预训练代码已开源在https://github.com/thu-pacman/AIPerf/tree/Atlas/Bert。

（2）DeepFM训练

DeepFM是由哈尔滨工业大学和华为诺亚实验室联合提出的一种基于深度学习的点击率预测模型。该模型由FM和Deep两个部分构成。FM部分通过因子分解的方式对于推荐特征进行分解和分析，Deep部分利用多层前馈网络来学习推荐特征之间的交互关系。两者并列进行，对推荐特征进行多维度的处理，从而获得更准确的点击率预测。该推荐模型使用Criteo Kaggle Display Advertising Challenge数据集进行训练，可以选择NVIDIA GPU平台或华为昇腾平台。在训练时，用户可以选取不同的batch size来提高硬件利用率，但不可改变模型的尺寸。以点击率预测精度达到一定值作为训练有效判定条件。以单位时间内训练的样本数作为性能评价指标。DeepFM训练代码已开源在https://github.com/thu-pacman/AIPerf/tree/Atlas/DeepFM。

## 七、评分规则与评测指标

本设计方案采用多评测指标，全面评测AI集群的综合性能。

AIPerf500目前的评价指标是AI ops，即平均每秒处理的混合精度AI浮点操作数。评测指标的主要计算方法是，统计在规定的评测时间内所需的AI操作数，然后除以所需评测时间，作为最后的评价指标。同时，搜索和训练的模型需要到达一定精度（默认为70%），评价指标才被视为有效。

AIPerf-MoE使用与AIPerf500类似的评分方式，在一次训练过程中，记录从预处理结束开始训练到最后一轮迭代完成的时间，并计算这些迭代中所有 Transformer Block 的总计算量，进而求出训练过程的Tops作为分数。此外，还需要满足此次训练的模型最终验证lm-loss不大于3.0，其结果才视为有效。

训练和推理的吞吐率也是需要测试的性能指标。在AIPerf 500的AutoML训练过程中，可以根据模型的训练时间和计算量求出，其计算方法为 训练吞吐率 = 。对于推理的吞吐率，我们采用AIPerf500的初始网络，即Resnet50，进行评测，其计算方法为 推理吞吐率 = 。

同时，本方案也参考了其他现有AI评测程序，将Resnet-50的top-1准确率训练至75.9%所需的时间纳入评测指标。在测试时，不可以使用任何预训练模型，但可以修改训练相关的参数，如batch size、learning rate等参数，以在相应加速卡上实现最高效的训练。

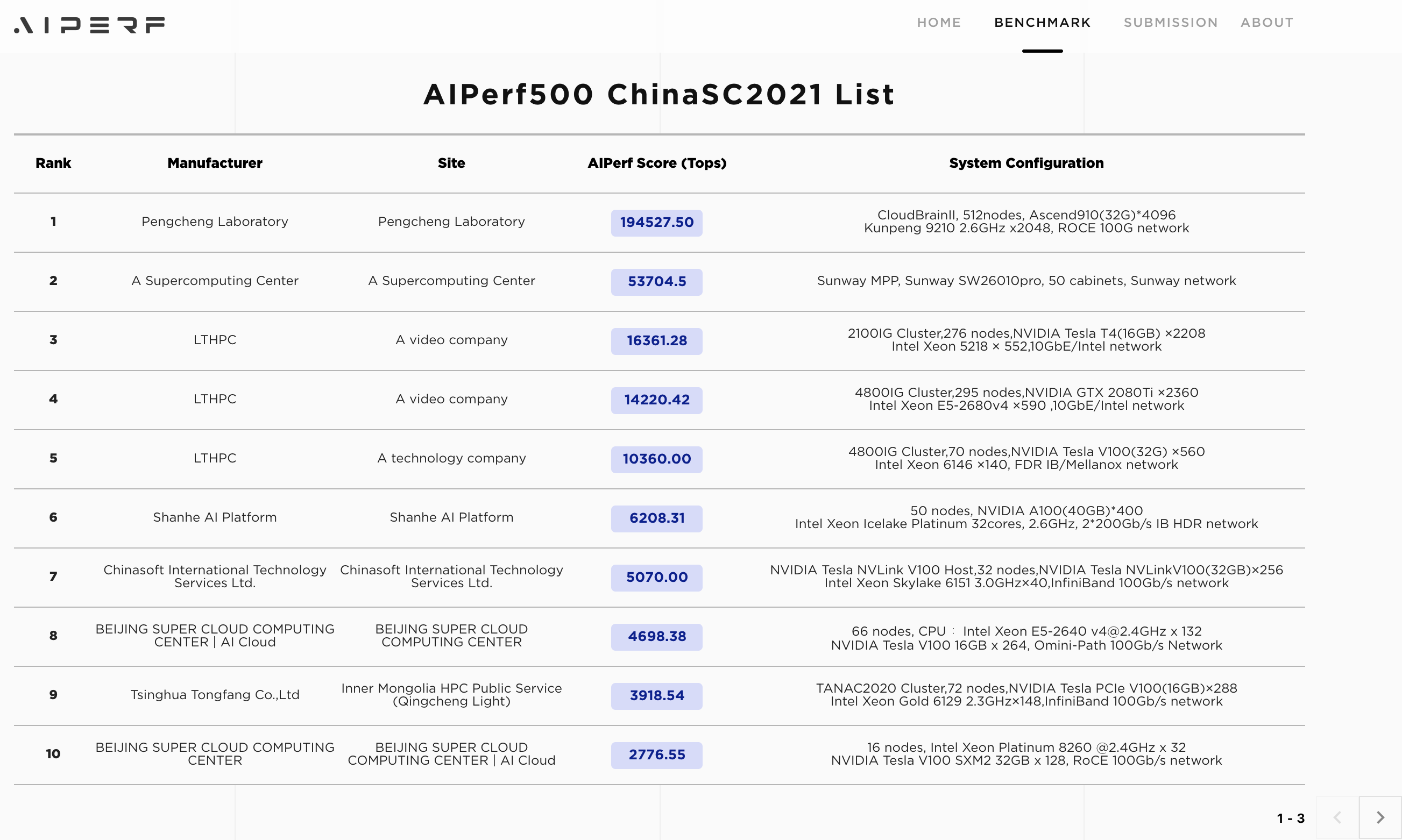
## 八、AIPerf相关工作——网站、榜单与论文

AIPerf网站已建成并开始使用（<https://aiperf.org/>），用户可以在网站上查看AIPerf相关论文、最新榜单、下载benchmark并提交结果。



AIPerf网站

2021年 11 月 12 日（中国，北京），第三届中国超级算力大会（ChinaSC2021）在北京顺利举行。会上由中科院计算所张云泉研究员，清华大学陈文光教授，美国脸书Pavan Balaji研究员和瑞士苏黎世联邦理工Torsten Hoefler教授联合ACM SIGHPC China委员会共同发起的基于AIPerf大规模人工智能算力基准评测程序的国际人工智能性能算力500排行榜发布。深圳鹏城实验室研制的基于ARM架构和华为处理器的鹏城云脑二主机以194527.5 Tops的AIPerf算力荣登榜首。



2021年AIPerf500 前十名

从本次发布的结果来看，AIPerf500基准测试程序已经可以有效的区分并评测从单机多卡的高性能计算节点到4000卡以上规模超级计算机的人工智能算力。硬件方面，AIPerf已支持Nvidia V100、A100、Ascend910、SW26010pro等多款国内外芯片及AI加速器。相关论文《AIPerf: Automated machine learning as an AI-HPC benchmark》已发表于Big Data Mining and Analytics （2021）。