OrionX AI 算力资源池化解决方案技术白皮书（社区版）

发布时间：2025 年 9 月

商标声明

本文档提及的商标为北京趋动科技有限公司的商标。

免责声明

本文档可能含有预测信息。由于实践中存在很多不确定因素，可能导致实际结果与预测信息存在差别。因此，本文档的信息仅供参考，不构成任何要约和承诺。趋动可能不经通知即修改本文档信息，恕不另行通知。

联系我们

电话：010-62560919

邮箱：BD@virtaitech.com

地址：北京市海淀区学清路甲 18 号中关村东升科技园学院园六层 A609 室

目录

图表目录

1 引言

人工智能作为引领新一轮科技革命和产业变革的颠覆性技术，已成为国际竞争的新焦点和经济发展的强大引擎。当前，我国人工智能与各行各业的协同发展日趋加速，“人工智能+”的概念深入人心。

2024 年，“人工智能+”行动首次被写入政府工作报告。2025 年政府工作报告提出，持续推进“人工智能+”行动。2025 年 8 月 26 日，“人工智能+”行动迎来重要进展：国务院印发的《关于深入实施“人工智能+”行动的意见》对外发布，明确了实施“人工智能+”行动的总体要求、发展目标和重点方向。

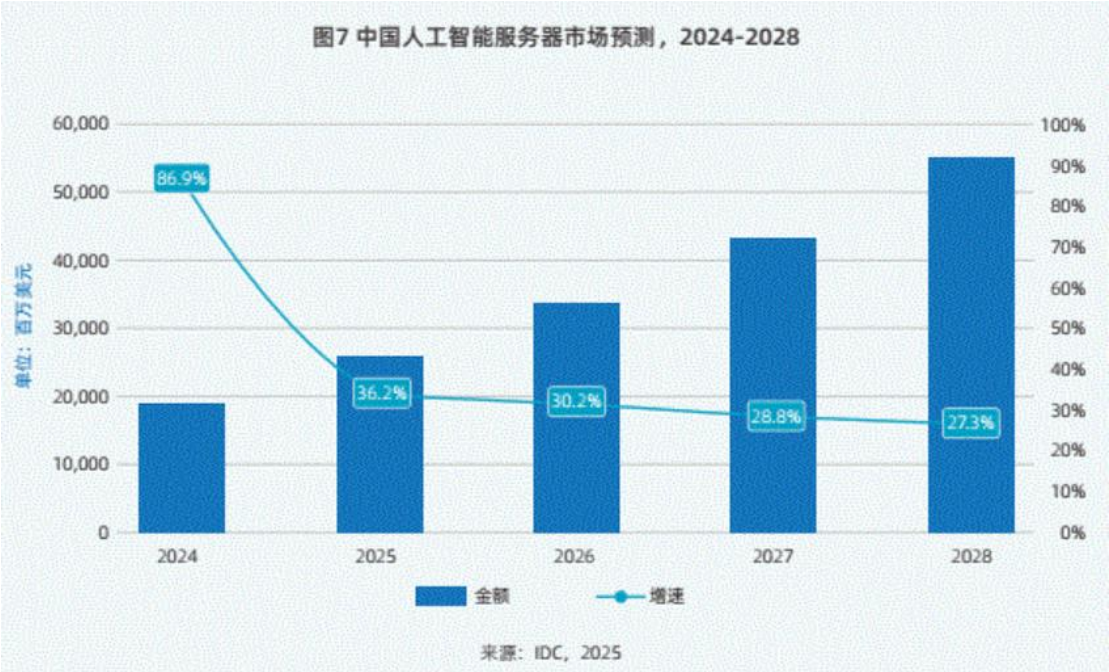
意见深刻把握人工智能技术和产业演进规律，明确了我国实施“人工智能+”行动的阶段性目标：到 2027 年，新一代智能终端、智能体等应用普及率超 70%，智能经济核心产业规模快速增长；到 2030 年，新一代智能终端、智能体等应用普及率超 90%，智能经济成为我国经济发展的重要增长极；到 2035 年，我国全面步入智能经济和智能社会发展新阶段。

IDC 在 2025 上半年发布的《2025 年中国人工智能计算力发展评估报告》的预测数据表明：2024 年，中国智能算力规模达 725.3EFLOPS，同比增长 74.1%，增幅是同期通用算力增幅(20.6%)的 3 倍以上；未来两年，中国智能算力仍将保持高速增长。

2025 年，中国智能算力规模将达到 1,037.3 EFLOPS，较 2024 年增长 43%；2026 年，中国智能算力规模将达到 1,460.3 EFLOPS，为 2024 年的两倍。

图表 1-1 中国智能算力和通用算力规模及预测

报告中还指出，2024 年中国人工智能算力市场规模达到 190 亿美元，2025 年将达到 259 亿美元，同比增长 36.2%，2028 年将达到 552 亿美元。



图表 1-2 中国人工智能服务器市场预测

伴随着人工智能产业的高速发展，对于 AI 算力的需求增长越来越快，与此同时，由于缺乏高效经济的 AI 算力资源池化解决方案，导致绝大部分企业只能独占式地使用昂贵的 AI 算力资源，带来居高不下的 AI 算力使用成本；由于缺少对异构算力硬件支持，用户不得不修改 AI 应用以适应不同厂商的 AI 算力硬件。这会加剧 AI 应用开发部署复杂性、提高 AI 算力投入成本并导致供应商锁定。

2 GPU 资源池化技术的演进

GPU 资源池化技术从初期的虚拟机 GPU 虚拟化，到多芯池化，经历了五个技术演进阶段。

- **虚拟机 GPU 虚拟化**。将物理 GPU 按照 2 的 N 次方，切分成多个固定大小的 vGPU（Virtual GPU，虚拟 GPU），每个 vGPU 的算力和显存相等。实践证明，不同的 AI 模型对于算力、显存资源的需求是不同的。所以，这样的切分方式，并不能满足 AI 模型多样化的需求。

- **容器 GPU 虚拟化**。将物理 GPU 按照算力和显存两个维度，自定义切分，获得满足 AI 应用个性化需求的 vGPU。

- **GPU over IP/IB**。AI 应用与物理 GPU 服务器分离部署，允许通过高性能网络远程调用 GPU 资源。这样可以实现 AI 应用与物理 GPU 资源剥离，AI 应用可以部署在私有云的任意位置，只需要网络可达，即可调用 GPU 资源。

- **GPU 池化**。形成 GPU 资源池后，需要统一的管理面来实现管理、监控、资源调度和资源回收等功能。同时，也需要提供北向 API，与数据中心级的资源调度平台对接，让用户在单一界面，就可以调度包括 vGPU 在内的数据中心内的各类资源。

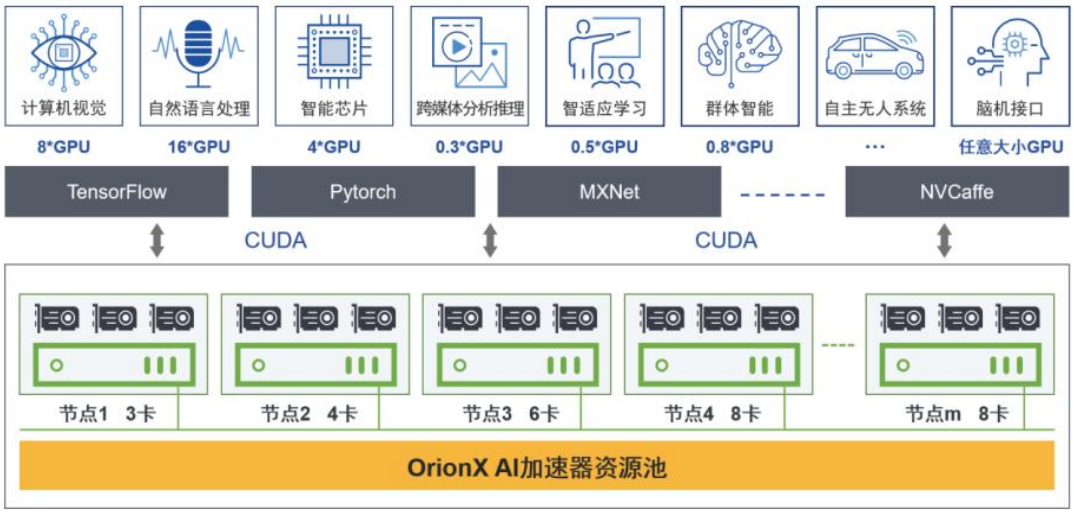
- **多芯池化。**不仅仅是单一厂商的 GPU，主流的国产 GPU 卡都能够实现统一的管理和调度，同时支持以不同的形态进行部署（虚拟机/容器/物理机/Kata Container 等）。

图表 2-1 GPU 资源池化技术演进图

3 OrionX 产品概述

趋动科技的 OrionX（猎户座）AI 算力资源池化解决方案已经实现了上述四个阶段的技术功能，可以为用户提供 GPU 资源池化的整体解决方案。

OrionX 帮助客户构建数据中心级 AI 算力资源池，使用户应用无需修改就能透明地共享和使用数据中心内任何服务器之上的 AI 加速器。OrionX 不但能够帮助用户提高 AI 算力资源利用率，而且可以极大便利用户 AI 应用的部署。



图表 3-1 OrionX **架构图**

OrionX 通过软件定义 AI 算力，颠覆了原有的 AI 应用直接调用物理 GPU 的架构，增加软件层，将 AI 应用与物理 GPU 解耦合。AI 应用调用逻辑的 vGPU，再由 OrionX 将 vGPU 需求匹配到具体的物理 GPU。OrionX 架构实现了 GPU 资源池化，让用户高效、智能、灵活地使用 GPU 资源，达到了降本增效的目的。

4 OrionX 产品优势

OrionX 通过构建 GPU 资源池，让企业内的 AI 用户共享数据中心内所有服务器上的 GPU 算力。AI 开发人员不必再关心底层资源状况，专注于更有价值的业务层面，让应用开发变得更加便 捷。OrionX 产品有如下优势：

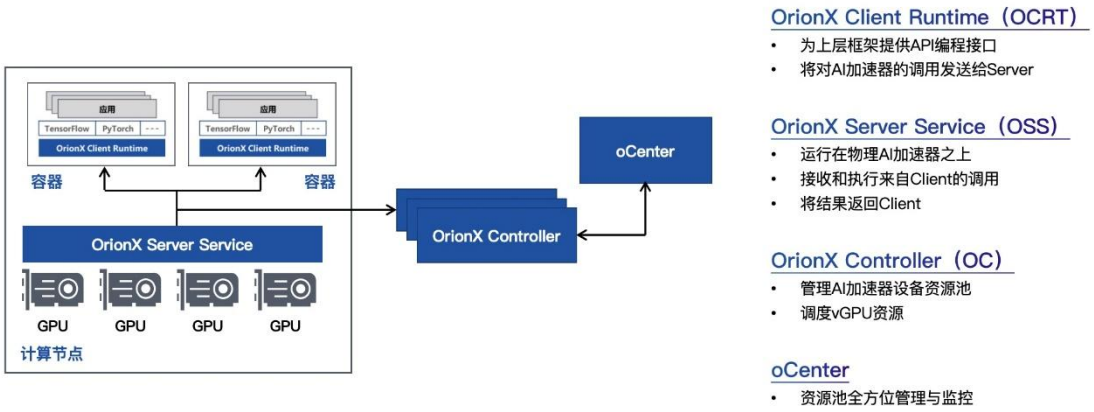
- **提高利用率**
    - 支持将 GPU 切片为任意大小的 vGPU，从而允许多 AI 负载并行运行，提高物理 GPU 利用率。
    - 提高 GPU 综合利用率多达 3-10 倍，1 张卡相当于起到 N 张卡的效果，真正做到昂贵算力平民化。
- **高性能**
    - 相比于物理 GPU，OrionX 本地 vGPU 性能损耗几乎为零。
    - vGPU 资源隔离，并行用户无资源互扰。
- **全局管理**
    - 提供 GPU 资源管理调度策略。
    - GPU 全局资源池性能监控，为运维人员提供直观的资源利用率等信息。

5 OrionX 软件架构

5.1 OrionX 的逻辑架构

一个典型的 OrionX GPU 资源池的逻辑架构中包含了 OrionX Controller（OC）、OrionX Server Service（OSS）、OrionX Client Runtime（OCRT）和 oCenter 等功能组件。

OrionX 的各功能组件可以根据用户环境需求被部署在单服务器上，也可以被分布式地部署在数据中心的多个物理机或者容器环境中。OrionX 的逻辑架构如下图所示。



图表 5-1 OrionX 逻辑架构图

CUDA(Compute Unified Device Architecture)是由 Nvidia 公司定义且公开推广、维护的一种 GPU 编程接口。从 2007 年推出之后，经过十几年生态培育，已经成为 GPU 编程的一个事实标准。大部分流行的 AI 框架，例如 TensorFlow、PyTorch、MXNet 和 PaddlePaddle 都是基于 CUDA 编程接口开发。

OrionX 在管理物理 GPU 之后，通过模拟 CUDA 标准接口，为各种 AI 应用提供一个与 Nvidia CUDA SDK 接口功能一致的运行环境，从而使得 AI 应用透明无感知地运行在 OrionX GPU 资源池之上。OrionX 不仅在单服务器上模拟了 CUDA 标准接口，并且通过分布式部署各功能组件，能够提供分布式的 CUDA 运行环境。

5.2 OrionX 的功能组件

5.2.1 OrionX Controller（OC）

OrionX Controller 是 GPU 资源池的核心管理调度模块，其他所有 OrionX 的功能组件都直接或者间接通过网络连接到 OrionX Controller, 并与其保持信息同步。为了实现 OrionX GPU 资源池的统一管理以及资源调度，节点 IP 地址、物理 GPU 信息、虚拟 GPU 信息以及应用任务信息等都会汇总至该组件。

一个 OrionX GPU 资源池可以只部署一个 OrionX Controller。OrionX Controller 提供如下功能：

- 各个功能组件的服务注册、服务发现功能。
- 弹性虚拟 GPU 的调度分配功能。
- License 管理。
- 提供运维所需要的各种 Rest API。

5.2.2 OrionX Server Service（OSS）

OrionX Server Service 发现并管理物理节点上的 GPU 资源，同时把物理 GPU 的计算能力通过 OrionX 的高性能私有协议提供给数据中心内的各个物理节点，以及各个物理节点上的容器。

OrionX Server Service 部署在 OrionX 资源池内的每一个 GPU 节点上。OrionX Server Service 提供如下功能：

- 发现和管理物理 GPU 资源。
- 把物理 GPU 资源抽象成弹性的 vGPU。
- 执行 AI 应用的 GPU 计算任务。
- 支持容器的网络隔离。

5.2.3 OrionX Client Runtime（OCRT）

OrionX Client Runtime 是一套兼容 Nvidia CUDA 编程环境的运行环境，模拟了 CUDA 的运行时接口。当 AI 应用在使用 Nvidia GPU 进行计算的时候，会自动调用 OrionX Client Runtime。由于 OrionX Client Runtime 提供和 Nvidia GPU 兼容的 CUDA 接口，因此应用无需修改，可以透明无感知地运行在一个虚拟的 GPU 环境下。

OrionX Client Runtime 部署在每一个应用环境下，替代原有的 Nvidia CUDA SDK。OrionX Client Runtime 提供如下功能：

- 兼容 CUDA 接口。
- 自动完成虚拟 GPU 资源的申请、释放等功能。
- 支持容器和宿主机的网络隔离。

5.2.4 OrionX GUI（OG）

OrionX GUI 给运维提供一个友好的 GUI 界面，方便管理员对 OrionX 整体资源池进行全面管理。OrionX GUI 提供如下功能：

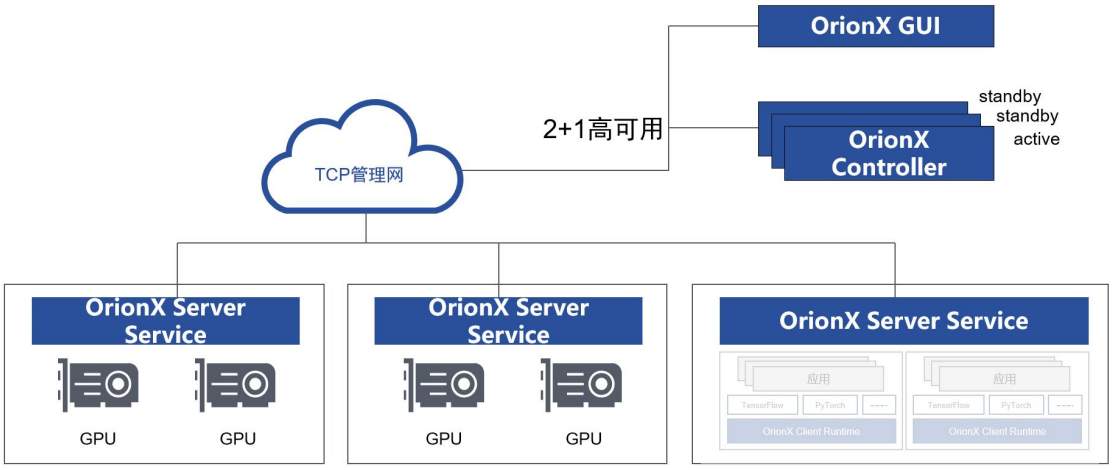
- 查看各组件的部署详情。
- 细粒度管理资源池内的资源。

5.3 OrionX 组件间通信

OrionX 的各个功能组件通过管理平面网络和数据平面网络进行通信，共同完成 GPU 资源池的管理以及 GPU 资源的调度等功能。

5.3.1 管理平面

在部署 OrionX 时，使用基于 TCP/IP 网络的管理平面，来承载整个系统的管理工作。通过管理网络，分布在各个节点的功能组件都保持和 OrionX Controller 同步。管理平面逻辑结构如下图所示。



图表 5-2 管理平面逻辑结构图

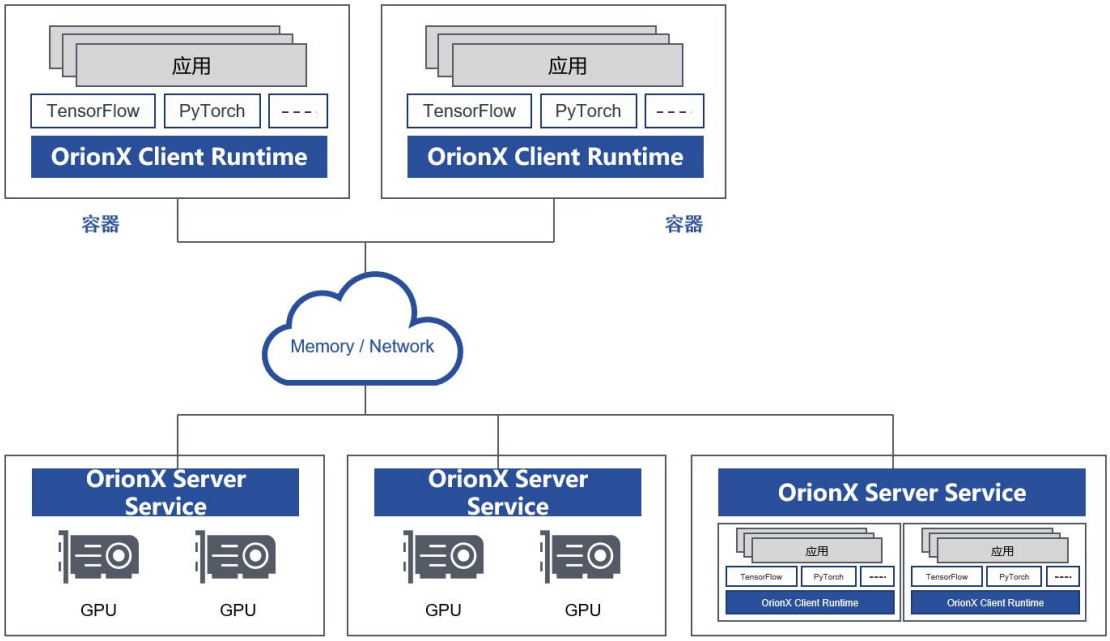
通过私有的同步协议，部署的各个功能组件具有如下特点：

- OrionX Controller 支持高可用的部署模式。
- 各个功能组件启动的次序无要求。
- 当某一个功能组件从错误中恢复之后，可以自动同步到正确的状态。

5.3.2 数据平面

在应用运行的过程中，应用所在环境和 GPU 物理节点之间的数据传输使用的是 OrionX 的数据面。该数据面支持多种后端数据传输载体，包括 TCP/IP 以太网络、RoCE RDMA、Infiniband RDMA、Share Memory 等。但社区版只支持 Share Memory，数据面具有如下的特点：

- 高带宽、低延迟。
- 同时支持多种传输协议，根据优先级自动使用高性能的传输方式。
- 支持容器和宿主机之间的 TCP/IP 网络隔离。



图表 5-3 数据平面逻辑结构图

6 部署形态

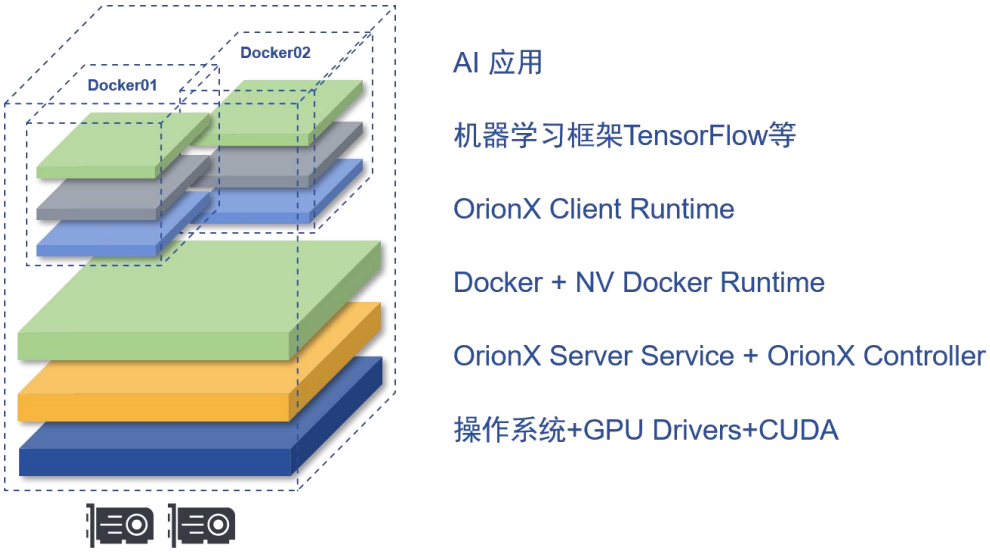
OrionX 的各个组件，支持直接部署在裸金属服务器上，即安装操作系统后，直接以 Binary 形式部署，也支持容器化部署。OrionX 具备适配多种 Linux 操作系统和云平台的能力，因此，OrionX 具有多样化的部署形式。

OrionX 支持 CentOS、Ubuntu、Debian 等 Linux 发行版本，同时支持基于 Docker 的容器云平台。尤其是支持原生容器，并实现了和 Kubernetes 的平滑对接。

6.1 OrionX 与容器云平台集成

OrionX 支持原生容器，各个组件都可以通过容器镜像方式部署。在容器环境中，客户只需要使用 OrionX 组件提供的启动脚本，就可以一键完成 OrionX 的组件安装，轻松实现 GPU 资源池化。

OrionX 的容器部署方式，将 GPU Drivers、CUDA、CUDNN 和 NCCL 等软件栈都下沉到宿主机上，容器内部只需要安装 OrionX Client Runtime 和机器学习框架，即可运行 AI 应用，



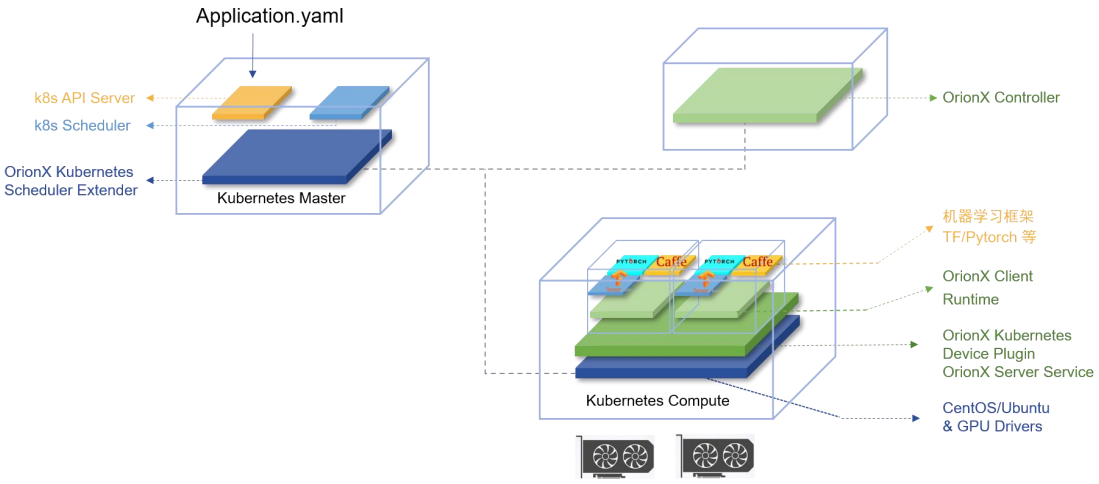
大大简化了客户算法工程师运维、管理 AI 基础架构的工作。

图表 6-1 OrionX 与容器云平台集成

6.2 OrionX 与 Kubernetes 集成

OrionX 为 Kubernetes 提供两个插件，实现与 K8S 的集成对接。集成后，系统管理员只需要在 K8S 中，即可完成对 GPU 资源池中 vGPU 资源的配置和调度管理。并且，允许系统管理员通过单一接口调度全部数据中心资源，实现 SDDC（Software Defined Data Center，软件定义的数据中心），这样就简化了运维工作。OrionX 为 Kubernetes 提供的两个插件是：

- **OrionX Kubernetes Device Plugin**
    - 通过和 OrionX Controller 通讯，获取 OrionX GPU 资源池信息。
    - 通过 Kubernetes 定义的 Device Plugin 标准向 Kubernetes 注册名字为 *virtaitech.com/gpu* 的资源。
- **OrionX Kubernetes Scheduler Extender**
    - 提供基于 HTTP API 通讯的松耦合调度扩展功能。
    - 通过配置文件向 Kubernetes 注册名字为 *virtaitech.com/gpu* 的资源敏感字，使其指向 Orion Kubernetes Scheduler Extender 的 HTTP 服务地址。



图表 6-2 OrionX 和 Kubernetes 集成

7 OrionX 应用场景

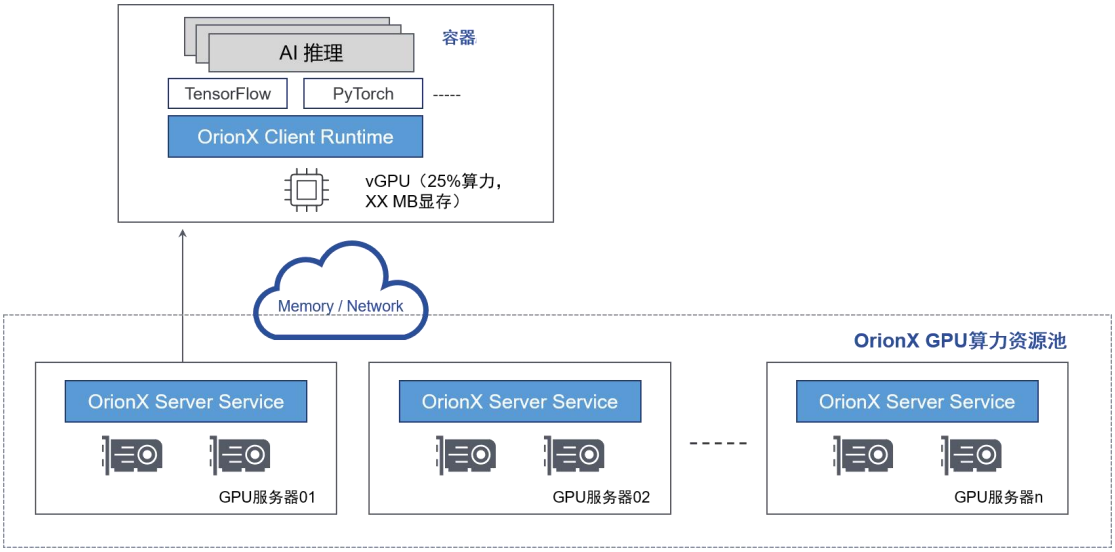
7.1 OrionX 支持小模型场景的典型应用

小模型如推理、开发、教学实训等场景，对算力资源需求量小，通常不能占满一张 GPU 卡资源。作为 AI 算力资源池平台，OrionX 可以从算力和显存两个维度，切分 GPU。支持将多个小模型任务调度到一张卡，有效提高资源利用率。

7.1.1 通过“化整为零”功能支持推理

OrionX 支持将一块物理 GPU 细粒度切分成多块 vGPU，然后分配给多个虚拟机或者容器。每一块 vGPU 的显存和算力都能被独立设置和限制。通过这个功能，用户可以高效地共享 GPU 资源，提高 GPU 利用率，降低成本。

算力切分的最小颗粒度为原物理 GPU 算力的 1%；显存切分的最小颗粒度为 1MB。



图表 7-1 通过化整为零功能支持推理

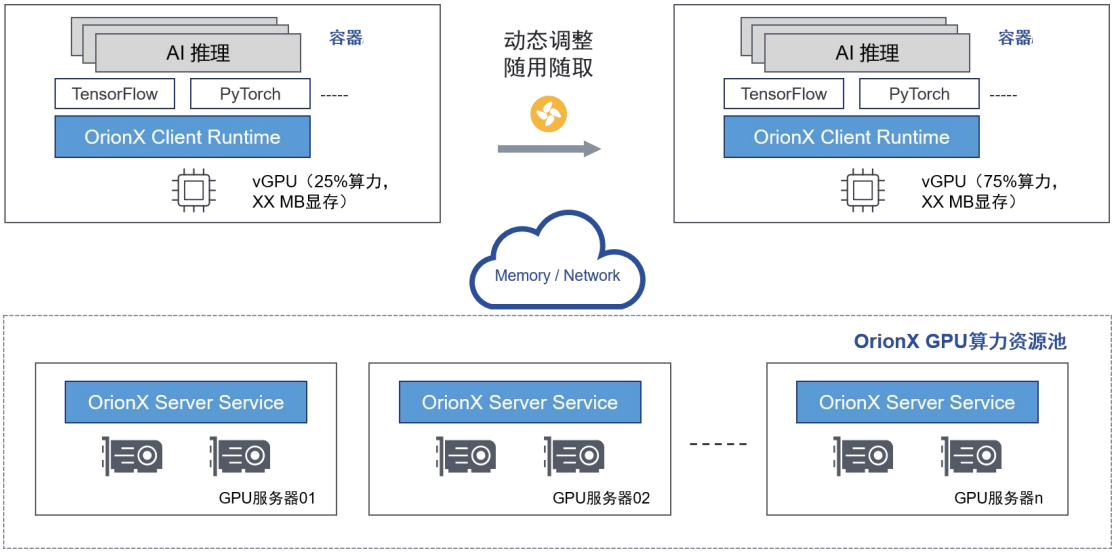
7.2 OrionX 支持大/小模型场景的典型应用

7.2.1 通过“随需应变”功能支持训练/推理

OrionX 支持用户在虚拟机或者容器的生命周期内，动态分配和释放所需要的 GPU 资源。通过这个功能，OrionX 帮助用户实现了真正的 GPU 资源动态伸缩，极大提升了 GPU 资源调度的灵活度。

OrionX 支持 vGPU 资源按需分配、随用随取，最大限度的利用算力资源。不论是大模型训练，还是小模型推理的环境中，用户都可以按照 AI 模型需求，动态的调整算力资源大小，而无需重启挂载 vGPU 的虚拟机/容器。OrionX 支持 vGPU 资源预留模式和获取模式：

- **预留模式**：和使用物理 GPU 类似，客户申请的 vGPU 是独占的，不可被其他用户使用。
- **获取模式**：客户申请的 vGPU 是动态的，只有在客户的 AI 应用运行时，vGPU 资源才锁定到具体的物理 GPU，一旦 AI 应用结束，物理 GPU 资源及时释放。



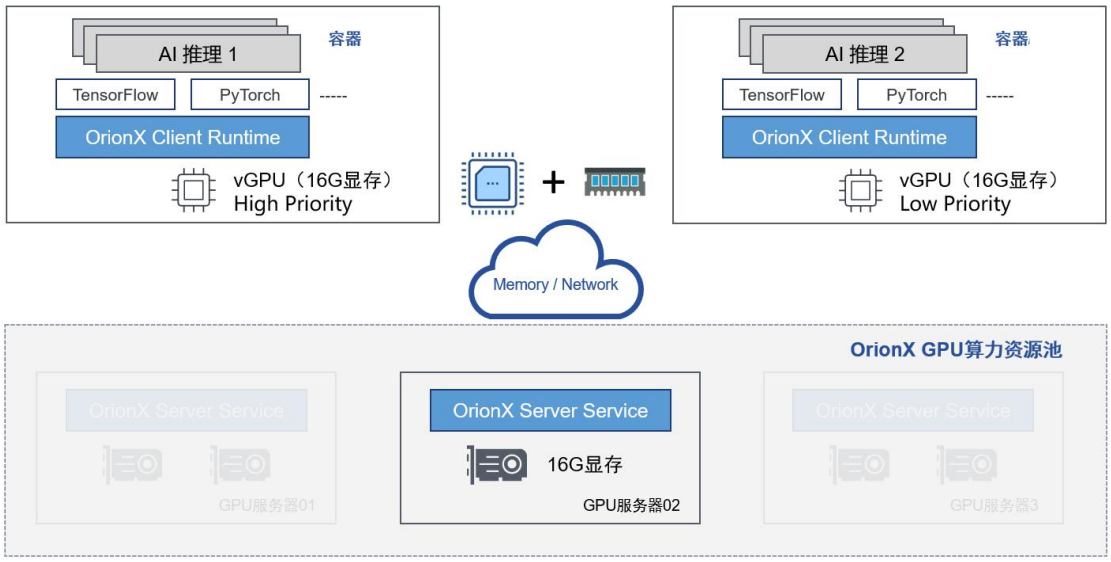
图表 7-2 通过随需应变功能支持训练/推理

7.2.2 通过“显存超分”功能支持多任务叠加常驻

通常推理任务为满足最佳用户体验，会将推理模型常驻显存，24 小时不中断，以便拥有最快响应速度。但是这类常驻任务一般算力利用极低，而且潮汐效应明显。

OrionX 支持多任务潮汐叠加。通过“显存超分”，OrionX 会调用系统内存补充 GPU 显存，在逻辑上扩大 GPU 显存的承载容量，从而支持多个常驻显存的长尾任务叠加在同一个物理 GPU 上，提高单个 GPU 的承载量，充分利用 GPU 闲置算力。

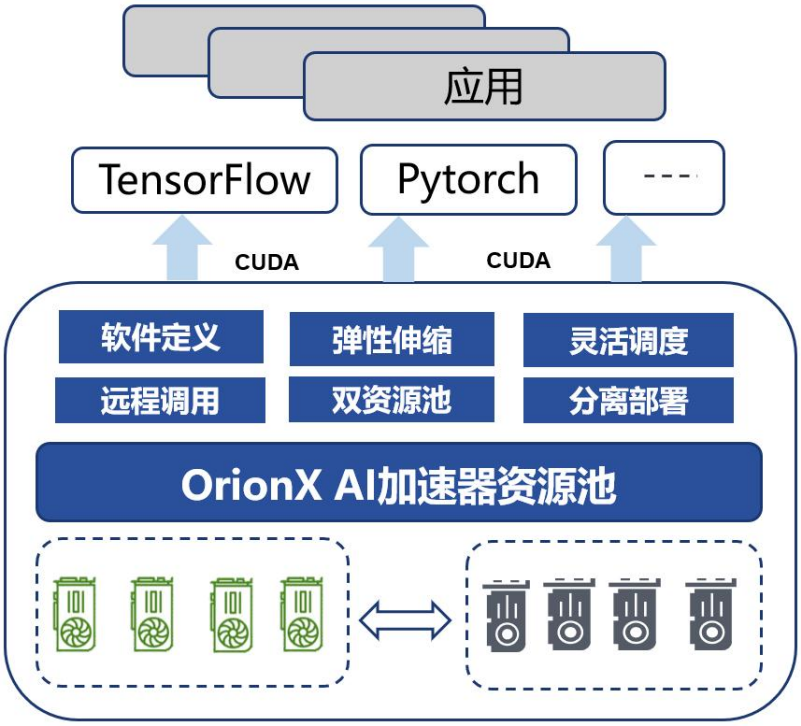
根据业务特点，OrionX 还支持不同任务设置不同优先级，从而保证突发高优先级任务的服务质量。



图表 7-3 通过显存超分功能支持多任务叠加常驻

7.2.3 通过“双类资源池”功能支持物理/虚拟切换

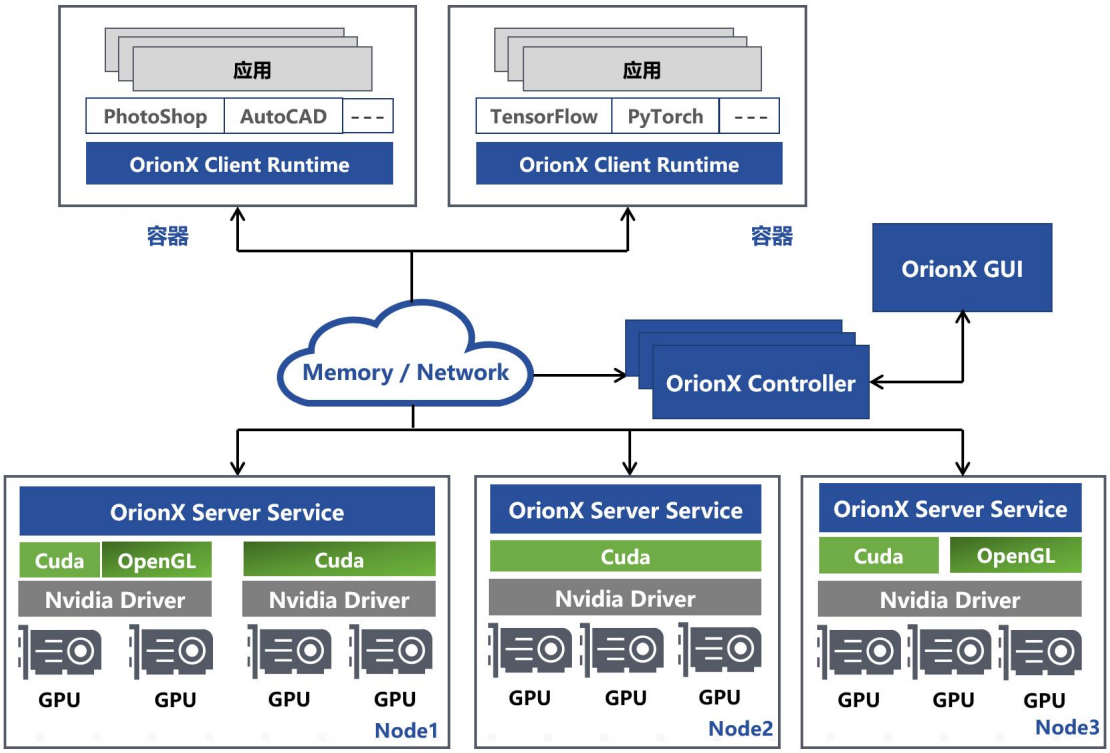
个别 AI 任务由于程序本身自有的特殊性，需要直接使用物理 Native GPU 资源，OrionX 支持同时纳管 OrionX GPU（即经过 OrionX 池化管理的 GPU，可以被虚拟化为多个 VGPU），和 Native GPU（即原生 GPU，不会被虚拟化）。OrionX 能够在一个界面上方便的控制哪些 GPU 卡初始化上报为 OrionX GPU，哪些 GPU 卡被初始化上报为 Native GPU。在初始化上报结束以后，依然能够灵活的在 OrionX GPU 和 Native GPU 之间安全的做切换。通过“双类资源池”功能，OrionX 可以同时支持 AI 任务申请 Native GPU 或者申请虚拟化之后的 OrionX GPU 两类不同资源，以应对不同任务资源申请的需求。



图表 7-4 通过双类资源池功能支持物理/虚拟切换

7.2.4 通过“多 Arch”架构可同时支持 AI 计算与图形渲染

OrionX 支持 Multi-Arch（多 Arch）架构，实现客户在单个容器或虚拟机中分配 vGPU 算力资源，既可以用于 AI CUDA 模型计算，又可以支持 OpenGL 图形渲染。有效支撑了数字场景业务模型计算和渲染同时需要需求。在多 Arch 架构下，单个容器或虚拟机可运行多协议（CUDA+OpenGL）。解决了之前 AI CUDA 模型计算和 OpenGL 图形渲染应用，需要部署在不同容器或虚拟机带来的不便。



图表 7-5 通过多 Arch 架构可同时支持 AI 计算与图形渲染

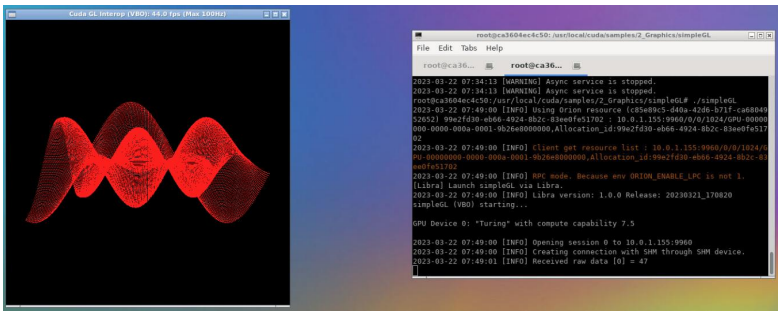
应用场景 1：虚拟数字人推理场景

虚拟数字人推理场景，目前已在金融等行业的虚拟 IP 形象代言、智能客服、营销等领域快速增长，为客户提供 24 小时不间断的服务，帮助企业降低人力成本，提升客户体验。业务容器和虚机申请的多 Arch vGPU 算力资源可以支持 AI 模型人物自然语言生成和人物动画生成渲染运行在同时运行。

图表 7-6 虚拟数字人推理场景

应用场景 2：研究科学计算场景

在大学等科研机构研发场景，如：人工智能、生物医药、粒子物理、程序化交易等领域，会使用 GPU 进行科学计算加速，过程中会使用 OpenGL 将模型计算结果渲染输出。通过多 Arch vGPU 算力资源可以支持模型计算和图形输出同时运行。



图表 7-7 研究科学计算场景

8 兼容性列表

- **CPU 架构**
  - X86（Intel/AMD）
  - C86（海光）
- **NVIDIA GPU**
  - NVIDIA Tesla 系列：H 系列、L 系列、A 系列、T 系列、V 系列、P 系列、M 系列
  - NVIDIA GeForce 系列：RTX 40 系列、RTX 30 系列、RTX 20 系列、GTX 10 系列、GTX 98/97/96/95/75 系列
  - NVIDIA Quadro 系列：RTX 系列、GV/GP 系列、P 系列、M 系列
- **NVIDIA CUDA**
  - CUDA 12.0，12.1，12.2，12.3，12.4，12.5，12.6
  - CUDA 11.0, 11.1, 11.2, 11.3, 11.4，11.5，11.6，11.7，11.8
  - CUDA 10.0, 10.1, 10.2
  - CUDA 9.0, 9.1, 9.2
- 操作系统
  - CentOS 7/8
  - Ubuntu 16 / 18 / 20
  - RHEL7/8
  - AlmaLinux 9.0
  - Rocky Linux 8.8
  - 麒麟 V10-SP1
- 云平台
  - 容器环境：Docker >= 19.0，containerd >=1.5
  - Kubernetes 环境：主流版本
- 深度学习框架
  - TensorFlow for CUDA : 1.8/ 1.9 / 1.10 / 1.11 / 1.12 / 1.13 / 1.14 /1.15 /2.0 /2.1 /2.2 /2.3 /2.4 /2.5 /2.6 /2.7/ 2.9/2.10 / 2.12/ 2.13/ 2.14/ 2.15
  - TensorRT : 5.x / 6.x / 7.0 / 7.1 / 7.2 /8.x
  - Pytorch for CUDA : 1.0 / 1.1 / 1.2 / 1.3 / 1.4 / 1.5 / 1.6 / 1.7 / 1.8/ 1.9 / 1.10/ 1.11/ 1.12/ 1.13/2.0/2.1/2.2
  - PaddlePaddle: 1.5 / 1.6 / 2.0 / 2.1 / 2.2/ 2.3.2
  - ONNX : 1.0 / 1.1 / 1.2 / 1.3 / 1.4 / 1.5 / 1.6 / 1.7 / 1.8 / 1.9 / 1.10 / 1.11/1.12

- MXNet :1.4.1 / 1.6 / 1.7 / 1.8 / 1.9
- XGBoost :0.72、0.8、0.9
- NVCaffe :1.0