
KFNative™ 晶格计算技术白皮书

生成日期：2026-09-19

第 1 章 背景与动机

1.1 AI 算力基础设施的演进趋势

计算体系正在经历结构性演化，方向可以概括为三个并行趋势：

1. **GPU/加速器成为算力主流**。随着 AI 大模型、HPC、大数据分析的普及，GPU、FPGA、DPU 等加速器从辅助部件上升为数据中心的核心计算资源，计算负载从通用 CPU 向加速器迁移。

2. **存储向高性能 SSD 迁移**。NVMe SSD 凭借低延迟、高并行度逐步替代 HDD 与 SATA SSD，成为企业级存储的主流介质。

3. **基础设施走向紧耦合（Converged / Tightly Knitted）**。为满足资源池化与共享需求，计算、存储、网络从"分布式/Scale-Out 的松耦合堆叠"走向"紧耦合融合"，以消除跨部件数据搬运的开销。

晶格计算（Lattice Computing）正是在这一背景下提出的**革命性全闪存储与算力融合技术**。

1.2 现有计算体系的三大痛点

KFNative™明确指出了当前计算体系中的三类痛点：

痛点	具体表现	影响
存储可靠性成本高	全闪存储已是千亿级市场规模，但高可靠性仍依赖价格昂贵的双端口 SSD	存储系统成本居高不下，规模扩张受限

容器云存储短板	Kubernetes 逐渐成为主流，但始终缺乏高性能、易部署的存储子系统	云原生应用难以获得原生 NVMe 级别的存储性能
GPU 算力被存储/网络钳制	GPU/AI 计算架构受限于存储性能，网络容易成为瓶颈	GPU 有效利用率低，算力无法充分释放

从互连协议视角看，传统 PCIe 架构虽然已成为事实上的高速串行总线标准（点对点拓扑、全双工链路，PCIe 5.0 单链路 32 GT/s、x16 双向各可达 64 GB/s），但其设计初衷是**批量数据传输**，并不提供跨设备的内存一致性与共享内存池管理机制（白皮书 CXL 章节明确指出现有 PCIe 中未规定一致性机制，无法有效管理隔离的内存池与多设备共享内存）。加速器访问主机内存通常依赖 DMA，主机无法缓存设备数据，导致 GPU 与 SSD 之间的大规模数据交换只能绕行网络或系统内存，产生毫秒级延迟。

1.3 KFNative™ 的总体路线

针对上述痛点，晶格计算提出"彻底解决带宽瓶颈、完全释放 **GPU** 算力"的技术路线：

- 以**晶格互联（Lattice Fabric）**即**延伸的 PCI Express（Extended PCIe）**作为统一数据平面，把 GPU pooling、NVMe SSD pooling、分布式 AI 训练、HPC、内存池化、大数据、VR/AR/渲染等负载统一承载在 PCIe 互联之上；
- 追求 **Flexible composition（灵活组合）**、**High performance（高性能）**、**High utilization（高利用率）** 三个设计目标；
- 通过 **GPU/SSD 紧耦合**，让数据在延伸的 PCIe 互联中完成端到端交换，将数据交换延迟从毫秒级降低到微秒级，从而解除存储与网络对 GPU 算力的钳制。

> **本章结论**：存储与算力之间的数据通路是当前 AI 基础设施的最大瓶颈。晶格计算以"紧耦合 + 延伸 PCIe 互联"为总体路线，目标是在保持标准硬件生态（PCIe/NVMe/SSD）的同时，从架构层面消除 GPU 与存储之间的延迟与带宽鸿沟。

第 2 章 核心技术原理与 KFNative™ 定位

2.1 KFNative™ 的核心算法思想：GPU/SSD 紧耦合

晶格计算的本质是一种**开创性的计算架构**：基于晶格互联（Extended PCIe Fabric）实现 **GPU 与 SSD 的紧耦合**，使 GPU 与 SSD 之间的数据交换**端到端完全在延伸的 PCIe 互联中完成**，不再绕行网卡、网络交换机与系统内存。该架构带来的直接收益是：

- **数据交换延迟数量级降低**：传统路径（GPU → PCIe → 网卡 → 网络 → 网卡 → PCIe → SSD）耗时毫秒级；晶格互联路径（GPU ↔ PCIe Switch ↔ SSD）耗时微秒级。
- **为 KV-Cache 等 AI/LLM 关键数据结构提供原生高速通路**：GPU 通过相邻的 SSD 交换数据，把大容量存储"贴"到 GPU 身边。

2.2 KFNative™ 的角色定位

KFNative™ 是晶格计算的技术品牌与核心技术内核。 KFNative™ 被定位为三层技术栈中的第一层：

层级	技术	定位
1	KFNative™（KV Cache、Vector Database）	核心技术引擎
2	Tier0 NFS / Native NVMe（VM/Kubernetes）	存储服务层
3	Lattice PCIE Fabric（晶格互联）	底层互联基座

配套的 KFNative™ 确认了该名称作为晶格计算的注册商标与产品品牌使用。

KFNative™ 的核心机制包括四要素：

4. **GPU/SSD 配对组合**：以 PCIe Switch 为枢纽，将 GPU 与 SSD 组成固定配对单元；
5. **每片 GPU 拥有专有的计算缓存与 KV-Cache**：KV-Cache（大模型推理中的键值缓存）不再依赖昂贵的 HBM 容量，而是落在 GPU 专有的近端 SSD 缓存上；
6. **GPU 通过相邻 SSD 交换数据**：数据交换只经过延伸的 PCIe 互联，路径短、无网络跳数；

7. 数十倍地增加 **GPU/KV-Cache 的有效带宽**：通过"邻近 + 并行"的物理拓扑，把单 GPU 可访问的缓存/存储带宽放大一个数量级以上。

2.3 KFNative™ Ring Extension：绕开 HBM 与 NVLink

KFNative™ 的 Ring Extension 机制是其在训练场景的核心突破，通过把 GPU 组织为环形互联（Ring）拓扑，实现：

- **Commodity GPU, less HBM dependency**：使用普通商用显卡 GPU，降低对高成本 HBM 显存的依赖；
- **No NVLink dependency**：完全不需要 NVIDIA NVLink 高速互联，摆脱厂商专有互连的绑定与成本；
- **Better parallelism, All/All Ring Reduce**：利用环形拓扑实现高效的 All/All Ring Reduce 集合通信，提升并行训练效率；
- **Train largest LLM with 64X fewer GPUs**：以少 64 倍的 GPU 数量训练最大规模的大语言模型。

这一机制的意义在于：**HBM 与 NVLink 是大规模 LLM 训练成本的两大来源**。KFNative™ 通过把 KV-Cache 与中间数据"流式化"到近端 SSD，把训练的内存墙与互连墙同时打破，使显卡级 GPU 也能被有效组织起来承担大规模训练。

2.4 推理流式计算示例（KFNative™ 工作原理的量化说明）

以 GPT-3 175B 模型推理为例给出了流式计算的量化示例（96 层 Transformer）：

参数	数值	说明
模型规模	GPT-3 175B，96 层	每层约 1.8B 参数，约 3.6 GB
存储带宽	Gen6 x16 SSD，约 100 GB/s	新一代 PCIe Gen6 SSD
单层模型加载时间	3.6 / 100 ≈ 35 ms	从 SSD 流式加载一层权重
单层计算时间	100K × 1.8B / 1000T ≈ 180 ms	每层前向计算的估算
单层总耗时	≈ 215 ms	加载与计算重叠形成流水线

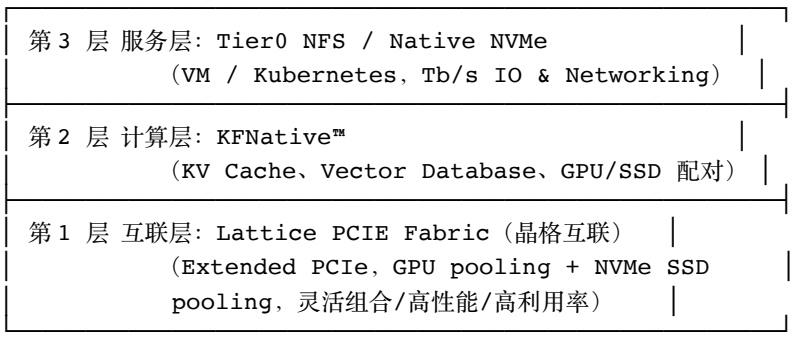
该示例展示了"**边加载、边计算**"的流式计算思想：推理引擎逐层从 SSD 读取权重，在权重到达的同时完成该层计算，加载时间（35 ms）远小于计算时间（180 ms），加载可以完全被计算隐藏。这使得"模型权重常驻 SSD、逐层流式进入 GPU"成为可行的推理范式，不再要求整个模型常驻显存。

> 本章结论：KFNative™ 是晶格计算的技术内核——以"GPU 邻接 SSD"的物理拓扑（配对组合 + Ring Extension）把 KV-Cache 与模型权重流式化，从根本上绕开 HBM 容量与 NVLink 互连的约束，以标准 PCIe/NVMe/SSD 组件实现数十倍的有效带宽放大与数量级的延迟改善。

第 3 章 系统架构、数据流与调度机制

3.1 系统架构分层

晶格计算系统从下至上由三层构成：



- **互联层（Lattice PCIE Fabric）**：以延伸的 PCI Express 作为统一互联基座，向下连接 GPU 池与 NVMe SSD 池，向上承载分布式 AI 训练、HPC 计算、内存池化、大数据、VR/AR/Rendering 等多种负载。设计目标为灵活组合、高性能、高利用率。
- **计算层（KFNative™）**：实现 GPU/SSD 配对组合、KV-Cache 与向量数据库（Vector Database）的原生存储，是晶格计算的核心引擎。
- **服务层（Tier0 NFS / Native NVMe）**：面向 VM 与 Kubernetes 提供 Tier0 级 NFS 与原生本地 NVMe 存储服务，达到 Tb/s 级 IO 与网络吞吐。

在系统集成层面，KFNative™ 可有效地与 DELL、HPE、Supermicro、Lenovo 等主流品牌服务器厂商的整机集成，表明其硬件适配遵循标准生态路线。

3.2 数据流路径对比：传统路径 vs KFNative™

明确对比了两条数据通路：

传统 GPU/SSD 数据交换路径（毫秒延迟）：

```
GPU → PCIe → 系统内存→ 网卡→ 网络→ 网卡→ 系统内存→ PCIe → SSD
```

数据从 GPU 出发，需要经过主机内存中转、两次网卡与网络交换机的搬运，跨域传输引入毫秒级延迟，网络成为瓶颈。

晶格互联数据交换路径（微秒延迟）：

```
GPU ↔ PCIe Switch ↔ SSD          （完全在 Extended PCIe 内完成）
```

GPU 与 SSD 通过 PCIe Switch 直接互连，端到端数据交换完全在延伸的 PCIe 互联中完成，不经过网络栈，延迟由毫秒级降至微秒级。

这一差异的核心在于：**晶格计算把存储"放进了 GPU 的邻居位置"**，用 PCIe Fabric 取代网络作为存储-算力数据平面，从而消除网络跳数与协议栈开销。

3.3 KV-Cache 调度与流式计算机制

3.3.1 推理场景：逐层流式加载

以 GPT-3 175B 推理为例，模型权重不常驻显存，而是常驻近端 SSD。调度机制按"**逐层流水线**"运转：

8. 推理引擎从 SSD 流式加载第 L 层权重（约 3.6 GB，耗时约 35 ms）；
9. 加载的同时，GPU 对第 L-1 层执行前向计算（约 180 ms）；
10. 加载时间远小于计算时间，加载完全被计算隐藏，形成连续流水线；
11. KV-Cache 以 Paged（分页）方式管理，落在每片 GPU 专有的计算缓存中，超限部分由相邻 SSD 承载。

3.3.2 训练场景：All/All Ring Reduce

KFNative™ Ring Extension 将 GPU 组织为环形拓扑，集合通信采用 All/All Ring Reduce：每片 GPU 只需与相邻节点交换数据，梯度与中间结果沿环流动汇聚，实现高并行度的规模化训练，且不依赖 NVLink。

3.3.3 调度特征总结

特征	说明
数据本地性	GPU 优先访问相邻 SSD，路径最短
流水线重叠	权重加载与计算重叠，隐藏存储延迟
集合通信	Ring 拓扑 All/All 归约，无 NVLink 依赖
存储虚拟化	KV-Cache/向量数据库原生落在 NVMe 池，Paged 管理

本章结论：晶格计算架构的本质是"把存储放进 GPU 的邻居"。互联层（Extended PCIe Fabric）提供统一数据平面，KFNative 计算层实现配对与流式调度，服务层面向 VM/Kubernetes 开放原生 NVMe 能力；数据流从"绕网络"变为"走 PCIe"，调度围绕逐层流式流水线与 Ring 归约展开。

第 4 章 硬件适配与互连技术协同

4.1 硬件适配策略：标准件与去特化

晶格计算的硬件适配遵循"使用标准生态组件（**Off-the-Shelf components**）"原则：

硬件	适配方式	说明
GPU	Commodity GPU（普通商用显卡）	不依赖特定厂商旗舰加速卡；降低对 HBM 的依赖（less HBM dependency），不依赖 NVLink（No NVLink dependency）
SSD	Gen6 x16 高性能 NVMe SSD（约 100 GB/s 带宽）	以新一代 PCIe Gen6 SSD 作为 KV-Cache 与权重流的近端载体
互连	PCIe Switch 组网	以标准 PCIe Switch 构建 Extended PCIe Fabric，连接 GPU 与 SSD

服务器	DELL / HPE / Supermicro / Lenovo 等品牌整机	KFNative™ 可与各品牌服务器厂商有效集成
CPU/内存	标准 x86 服务器内存体系	系统内存与 DDR5 生态兼容，非特殊硬件

这一"去特化"策略意味着晶格计算可以**利用任何厂商的服务器**、标准 NVMe SSD 与商用 GPU 搭建，从而显著降低系统成本与供应链风险。

4.2 与 PCIe 互连技术的协同

晶格互联的本质是**延伸的 PCI Express（Extended PCIe）**，因此 PCIe 是其底层基座：

- **物理层复用**：PCIe 是事实上的高速串行总线标准（点对点拓扑、全双工 lane 链路，PCIe 5.0 单链路 32 GT/s，x16 配置每方向可达 64 GB/s；Gen6 64 GT/s 采用 PAM-4 信令，x16 每方向约 128 GB/s），晶格互联直接构建于该物理层之上，天然继承其成熟生态与高带宽（PCIe 代际速率与分层模型详见 5.1 节）。
- **扩展路径**：传统 PCIe 面向端点互连，晶格计算将其"延伸"为 Fabric——通过 PCIe Switch 将 GPU、SSD 组织成网格状互联拓扑，实现 GPU pooling 与 NVMe SSD pooling。
- **与 PCIe 6.0 演进衔接**：PCIe 每代速率翻倍（Gen4 16 GT/s → Gen5 32 GT/s → Gen6 64 GT/s），晶格互联的带宽上限随 PCIe 代际提升而同步增长，其推理示例即基于 Gen6 x16 SSD（约 100 GB/s）推算。

4.3 与 CXL 互连技术的协同

CXL（Compute Express Link）是基于 PCIe 物理层的缓存一致性开放互连标准，与晶格计算形成互补而非竞争关系：

- **CXL 的核心能力**：通过 CXL.io / CXL.cache / CXL.mem 三协议在 CPU 与设备间建立缓存一致性的共享内存空间（数据以 528-bit FLIT 传输；CXL 1.x/2.0 基于 PCIe 5.0 PHY、32 GT/s 下 x16 每方向约 63 GB/s，CXL 3.0 基于 PCIe 6.0 PHY、64 GT/s 下 x16 每方向约 121 GB/s）；CXL 2.0 引入交换（Switch）与内存池化、持久内存支持；CXL 3.0 进一步支持 Fabric 级拓扑与多级交换。Type 1/2/3 设备分别对应无本地内存加速器、带本地存储/内存设备、纯内存扩展器（协议细节与设备类型详见 5.2 节）。

- **协同点一（内存池化方向一致）**：白皮书指出 CXL 通过内存池化满足超大规模数据中心高容量需求、降低软件栈复杂度；晶格计算的"内存池化（Memory pooling）"与"存储池化"同属资源池化趋势，两者在架构哲学上一脉相承。

- **协同点二（一致性与本地化的分工）**：CXL 解决"CPU 与设备内存的一致性共享"，晶格计算解决"GPU 与 SSD 之间的低延迟数据通路"；前者面向内存语义的细粒度一致访问，后者面向 KV-Cache/权重的流式大数据交换，可叠加部署。

- **协同点三（消除专有互连）**：CXL 与晶格计算都致力于减少专有内存互连（如 NVLink）的依赖，推动开放、异构、低成本的互连体系。

4.4 与 NVMe 及 DDR5 的协同

- **NVMe（存储接口）**：NVMe 是基于 PCIe 总线的非易失存储接口协议，支持多达 65,535 个队列（每队列 65,536 条命令）与高并行度，延迟远低于传统 AHCI/SATA（命令寄存器访问由 6–9 次降至 2 次、中断扩展至 2048 个 MSI-X）。晶格计算的 Native NVMe 存储服务直接利用 NVMe 的高队列并行度，实现 Tb/s 级 IO；其架构路线强调 **Native PCIE / 原生本地 NVMe**，区别于 NVMe over Fabric（NVMe-oF 经网络搬运）的间接路径（协议细节与测试要点详见 5.3 节）。

- **DDR5（主存）**：DDR5 作为主流系统内存技术（PC5-4800/5200/5600，对应 38.4–51.2 GB/s 带宽，1.1 V、32 banks、双 I/O 子通道）提供更高带宽与容量，构成服务器内存底座。晶格计算将**大数据量、流式数据**卸载到近端 NVMe SSD，系统内存/DDR5 承载 KV-Cache 的热点部分与计算过程中的临时数据，形成"**DDR5 热层 + NVMe SSD 冷/温层**"的分层缓存体系（DDR5 规格与测试工具详见 5.4 节）。

4.5 协同关系小结

互连技术	在晶格计算中的角色	协同价值
PCIe	Extended Fabric 物理基座	高带宽、成熟生态、代际演进
CXL	内存池化与一致性补充	与资源池化方向一致，可叠加部署
NVMe	原生本地存储接口	多队列高并行，Tb/s IO
DDR5	主存热层	与 SSD 构成分层缓存体系

本章结论：晶格计算建立在标准 PCIe/NVMe 生态之上，以商用 GPU 与 Gen6 NVMe SSD 为硬件底座；与 CXL 的内存池化/一致性方向互补协同，通过"去特化 + 标准件"显著降低系统成本与供应链依赖。

第 5 章 高速互连与存储技术详析：PCIe 5.0 / CXL / NVMe / DDR5

本章内容作为晶格计算底层互连与存储技术基座的技术详析。四类技术共同构成 KFNative™ Extended PCIe Fabric、原生 NVMe 服务层与 DDR5 主存热层的标准生态底座。

5.1 PCIe 5.0 技术详析

5.1.1 协议定位与分层模型

PCI Express（PCIe）是事实上的高速串行计算机扩展总线标准，采用**点对点拓扑**、**全双工 lane 链路**结构。PCIe 协议栈分为三层：

层次	职责
事务层（Transaction Layer）	生成/解析 TLP（事务层包），完成读写、完成、消息等事务语义
数据链路层（Data Link Layer）	通过 DLLP（数据链路层包）保证 TLP 可靠传输、流控、ACK/NAK 重传
物理层（Physical Layer）	电气信号、lane 初始化与训练、编码（8b/10b、128b/130b、PAM-4）

5.1.2 代际速率与通道配置

PCIe 每代速率遵循"约翻倍"演进规律，lane 宽度支持 x1 / x4 / x8 / x16 等配置：

代际	单 lane 速率	编码	典型 x16 单方向带宽
Gen1	2.5 GT/s	8b/10b	约 2 GB/s
Gen2	5 GT/s	8b/10b	约 4 GB/s
Gen3	8 GT/s	128b/130b	约 8 GB/s
Gen4	16 GT/s	128b/130b	约 16 GB/s

Gen5	32 GT/s	128b/130b	约 63–64 GB/s（每方向）
Gen6	64 GT/s	PAM-4	约 128 GB/s（每方向）
Gen7	128 GT/s（规划）	PAM-4	约 256 GB/s（每方向）

要点：

- **PCIe 5.0**：2019 年 5 月由 PCI-SIG 发布，速率 32 GT/s，x16 链路每方向最高 64 GB/s；首次引入"备用协议（Alternate Protocol）"部署能力，为 CXL 复用物理层奠定基础。首次量产落地包括 Intel 第 12 代酷睿（2021 年）、NVIDIA Hopper GPU（2022 年）与 AMD Zen 4 EPYC（Genoa，128 条 PCIe 5.0 lane）。

- **PCIe 6.0**：2022 年 1 月发布，64 GT/s，改用 **PAM-4 信令 + 前向纠错（FEC）**，x16 每方向约 128 GB/s。

- **PCIe 7.0**：2022 年 6 月由 PCI-SIG 宣布规划，目标 128 GT/s。

5.1.3 衍生接口与形态因子

接口	说明
OCuLink	面向外置 PCIe 的小型线缆接口，最高支持 PCIe 4.0 x4，常用于外接 GPU/存储扩展
M.2	面向消费/客户端 NVMe SSD，最高 PCIe 4.0 x4（新平台延伸至 Gen5）
U.2（SFF-8639）	2.5 英寸企业级 NVMe SSD 接口，最高 x4 PCIe lane，单服务器可组合多达 48 盘
U.3（SFF-TA-1001）	U.2 演进，"三模"（SAS/SATA/NVMe）单控制器，支持热插拔，向后兼容 U.2
EDSFF（E1.S/E3.S）	企业和数据中心标准形态因子，专为高密度 NVMe 设计，散热与供电优化

5.1.4 Gen5 测试环境与工具要点

- **Gen5 Switch** 卡（SerialCables，基于 Broadcom PEX89000）：在量产 Gen5 服务器稀缺阶段是搭建 Gen5 测试环境的唯一成熟选择；上行 Gen5 x16 金手指，下行提供 1 个 Gen5 x16 插槽 + 4 个 Gen5 x4 MCIO 接口（可经 MCIO 转 U.2/U.3/EDSFF 线缆测各类 Gen5 NVMe SSD；4 个端口可动态聚合为 x8/x16）。
- **PCIe Gen5 Retimer** 卡：协议感知的信号增强器件，用于补偿长走线/连接器插入损耗，重建 Gen5 32 GT/s 信号完整性。
- **Gen5** 延长线缆：85 ohm 阻抗、0.4–1 米、插入损耗约 4.5 dB/m，配合 MCIO 定义连接器。
- **JBOF（Just a Bunch of Flash）**：被动/主动盘柜形态，用于把多块 Gen5 SSD 从主机扇出。
- **Gen5** 协议分析仪（SerialTek KODIAK Gen5 x16）：支持 32 GT/s 线速抓包，144 GB 捕获缓冲、4 TB 存储、12 核 CPU，配合 SI-Fi 免校准眼图测量；亦可使用 BusXpert 软件工具链；Broadcom PEX89000 Switch 内嵌 PEA（协议事件分析）引擎。

5.1.5 Gen5 NVMe SSD 实测规格

产品	规格要点	实测/宣称性能
Kioxia CD8	PCIe Gen5 x4，U.2 2.5"，BiCS TLC；CD8-R（读密集 1 DWPD）7.68/15.3 TB，CD8-V（混合 3 DWPD）6.4/12.8 TB	128 GT/s（Gen5 x4）
Kioxia CM7	PCIe Gen5 x4，U.2 / E3.S；CM7-R 最高 30.72 TB（1 DWPD）、CM7-V 最高 12.8 TB（3 DWPD）；支持 SR-IOV、CMB、多流写入	AMD EPYC Genoa 平台实测顺序读最高 13.7 GB/s，较上代 CM6 顺序读提升 98.5%、顺序写提升 57%
Samsung PM1743	PCIe Gen5，U.2 / E3.S / AIC，7.68 / 15.36 TB	实测约 13.8 GB/s（Gen5 转接卡 + 12 代酷睿平台）

5.1.6 与晶格计算的关联点

晶格互联的本质是"**延伸的 PCI Express（Extended PCIe）**"：PCIe 5.0 提供单链路 32 GT/s、x16 每方向 64 GB/s 的物理基座，Gen6 64 GT/s 使带宽上限继续翻倍。KFNative™ 的推理示例即基于 Gen6 x16 SSD（约 100 GB/s）推算；GPU 与相邻 SSD 之间的数据交换端到端在延伸 PCIe 互联中完成，将传统"PCIe→网络→PCIe"的毫秒级路径压缩为微秒级。

5.2 CXL 技术详析

5.2.1 协议定位与历史

CXL（Compute Express Link）是面向高性能数据中心的**缓存一致性（cache-coherent）开放互连标准**，用于 CPU 到设备、CPU 到内存的高速连接。CXL 构建于 **PCIe 5.0 物理与电气接口之上**，包含基于 PCIe 的块 I/O 协议（CXL.io）以及两个新的缓存一致性协议（CXL.cache、CXL.mem）。

- 2019 年 3 月由 **Intel** 主导提出，CXL 联盟由九家创始成员（阿里巴巴、Cisco、Dell EMC、Meta/Facebook、Google、HPE、华为、Intel、Microsoft）于 2019 年 3 月组建、2019 年 9 月正式注册成立。
- 2020 年 4 月与 Gen-Z 联盟达成互操作合作，2021 年 11 月 **Gen-Z 规范并入 CXL**；2022 年 8 月 **OpenCAPI 规范并入 CXL**，统一了 CCIX、Gen-Z、OpenCAPI 等一致性互连路线。
- 2022 年 1 月起 AMD、NVIDIA、Samsung、Xilinx 进入董事会；ARM、Broadcom、IBM、Kioxia、Marvell、Micron、Qualcomm、SK Hynix、Synopsys、Western Digital 等为主要成员。

5.2.2 规范版本与速率

版本	发布时间	关键特性	速率（基于 **PCIe PHY**）
CXL 1.0	2019-03	基于 PCIe 5.0；CPU 可缓存一致性访问加速器共享内存	32 GT/s：x1 3.938 GB/s，x16 63.015 GB/s
CXL 1.1	2019-06	合规性与互操作性改进，向后兼容 1.0	同上
CXL 2.0	2020-11	支持交换（Switching）：扇出多设备、资源池化（pooling）、持久内存支持；链路级完整性加密（CXL IDE）；标准化 Fabric Manager；带宽无提升（仍用 PCIe 5.0 PHY）	同上
CXL 3.0	2022-08	基于 PCIe 6.0 PHY + PAM-4，带宽翻倍；多级交换与 fabric 能力（mesh/ring/ spline/leaf 拓扑）；增强一致性（peer-to-peer DMA、内存共享而非仅池化）	64 GT/s：x1 7.563 GB/s，x16 121.0 GB/s

CXL 端口采用 **Flex Bus**，同一端口可协商运行在标准 PCIe 5.0 模式或 CXL 模式，两类设备均可达 32 GT/s（x16 每方向最高 64 GB/s）。

5.2.3 三子协议与传输机制

子协议	功能	说明
CXL.io	基于 PCIe 5.0 的增强协议	承担配置、链路初始化与管理、设备发现与枚举、中断、DMA、寄存器 I/O（非一致性 load/store）；所有 CXL 设备必须支持
CXL.cache	设备访问/缓存主机内存	设备以请求-响应方式低延迟缓存主机内存；D2H 与 H2D 两个方向各含请求、响应、数据通道
CXL.mem	主机访问设备内存	主机（CXL.mem Master）以 load/store 访问设备（Subordinate）的易失性内存（RAM）或持久内存；M2S 事务含 Req/RwD，S2M 事务含 NDR（Cmp/Cmp-S/Cmp-E）/ DRS（MemData）

传输细节：

- 数据以 **528-bit（66 字节）FLIT**（Flow Control Unit）为单位在 PCIe 5.0 PHY 上传送，由 4 个 16 字节数据槽 + 2 字节 CRC 构成；FLIT 封装标准 PCIe TLP/DLLP 数据。**CXL 3.0** 在 PAM-4 模式下引入 **256** 字节 **FLIT**。

- CXL.cache 与 CXL.mem 共享同一套链路层/事务层，CXL.io 拥有独立链路层/事务层，二者经 **ARB/MUX（仲裁与复用）** 交织后走统一 PCIe 物理层；物理层含逻辑子层（可在 PCIe/CXL 模式间切换）与电气子层（遵循 PCIe 规范）。

- CXL.cache 的 D2H 请求语义包括 read / read0 / read0-write / write；H2D 监听（snoop）含 **SnpData**、**SnpInv**、**SnpCurr**；H2D 响应以 GO（Global Observation）编码的 MESI 状态管理缓存行，例如 DirtyEvict 逐出 64 字节修改行、RspIHitSE 表示共享/独占命中并使行失效。

5.2.4 设备类型与一致性模型

类型	协议组合	典型设备	说明
Type 1	CXL.io + CXL.cache	智能网卡等无本地内存加速器	依赖对主机 CPU 内存的一致性访问
Type 2	CXL.io + CXL.cache + CXL.mem	GPU、ASIC、FPGA（带 GDDR/HBM 本地内存）	CPU 与设备内存双向一致性共享；设备偏置（device bias）/ 主机偏置（host bias） 两种一致性模式按 4 KB 页粒度切换
Type 3	CXL.io + CXL.mem	内存扩展板、持久内存	向 CPU 提供低延迟的本地 DRAM 或字节寻址非易失存储

- Type 2 设备在**设备偏置模式**下可无事务直接访问本地内存（主机未缓存其行），在**主机偏置模式**下由主机缓存控制器处理全部访存；偏置状态存于设备本地翻译表中。CXL 3.0 以"增强一致性语义"取代偏置模式，允许设备修改本地内存后回写并使主机缓存失效，同时支持同一一致性域内设备的 peer-to-peer 传输与多设备共享同一内存段（区别于逐个分段的池化）。
- CXL 3.0 支持每个 root port 挂接多个 Type 1/2 设备、多级交换、**GFAM（Global Fabric Attached Memory）** 模式（内存设备经交换节点接入而无需直连主机），采用 **PBR（Port Based Routing）** 寻址，最多 4096 个节点。

5.2.5 实施与落地进展

- Intel Agilex FPGA 家族（2019 年 4 月）率先支持 CXL。
- **Samsung 128 GB DDR5 基 CXL 内存扩展模块**（2021 年 5 月），面向 TB 级内存扩展；2022 年 5 月发布基于自有内存控制器的 **512 GB CXL 内存模块**。
- 2021 年 Intel **Sapphire Rapids** 与 AMD **Zen 4 EPYC（Genoa/Bergamo）** 宣布支持 CXL 1.1。

5.2.6 对存储/计算架构的影响与晶格计算关联

- 白皮书指出：CXL 的一致性使内存池保持数据有效性，异构处理器（不同代际 CPU/加速器）可被统一纳管，低延迟被视为下一代用例与 PCIe 6.0 采用的关键使能；CXL 内存池化扩大了缓存内存池规模，契合超大规模数据中心"**资源解耦 + 开放架构**"趋势，并减少对专有内存互连的依赖。

- CXL vs CCIX：两者均基于 PCIe 5.0 物理层；CCIX 采用 peer-to-peer 对称一致性并基于 NUMA 池化，CXL 默认由 CPU 承担缓存一致性（非对称），带来可测的速度与延迟优势。

- **与晶格计算关联**：CXL 的"内存池化"与 KFNative 的"存储池化/资源池化"同属解耦池化趋势；CXL Type 2 的 GPU 一致性内存模型与 KFNative 的 KV-Cache 紧耦合方案可叠加部署——CXL 解决 CPU↔设备内存语义的一致性共享，晶格计算解决 GPU↔SSD 的流式大数据低延迟通路；两者均致力于消除 NVLink 等专有互连依赖，推动开放、异构、低成本互连体系。

5.3 NVMe 技术详析

5.3.1 协议定位与历史

NVMe（NVM Express / NVMHCIS）是开放的**逻辑设备接口**规范，通过 PCIe 总线访问非易失存储介质，专为利用 SSD 的低延迟与**内部并行性**设计。NVMe 逻辑由与存储介质共置的 NVMe 控制器芯片承载，规范版本演进（1.3→1.4→2.0）随介质侧更新，不影响主板/CPU 等 PCIe 兼容组件。

时间	里程碑
2007	Intel Developer Forum 提出 NVMHCI（NVM Host Controller Interface）
2008-04	NVMHCI 1.0 完成并发布
2011-03-01	NVMe 1.0 发布（NVM Express 工作组，90+ 公司参与，主席 Amber Huffman）
2012-10-11	NVMe 1.1：多路径 I/O（namespace 共享）、任意长度 scatter-gather I/O
2013-01	1.0e 更新；2013-07 Samsung XS1715 首款 NVMe 企业盘（3 GB/s 读，较前代快 6 倍）

2014-03	工作组注册为 NVM Express, Inc.（13 人董事会：Cisco、Dell、EMC、HGST、Intel、Micron、Microsoft、NetApp、Oracle、PMC、Samsung、SanDisk、Seagate）
2016	NVMe 1.2 引入 Host Memory Buffer（HMB），DRAM-less SSD 借用主机 DRAM 提升性能；NVMe-oF 1.0 标准发布
2020	NVMe 1.4a；之后进入重新架构的 NVMe 2.0 规范库体系

5.3.2 与 AHCI 的关键差异

维度	**AHCI**	**NVMe**
最大队列深度	1 个命令队列，每队列最多 32 命令	最多 65,535 个队列，每队列最多 65,536 条命令
非缓存寄存器访问（约 2000 周期/次）	非排队命令最多 6 次、排队命令最多 9 次	每条命令最多 2 次
中断	单中断	最多 2048 个 MSI-X 中断
多线程并行	发命令需同步锁	无锁
4 KB 命令效率	命令参数需 2 次串行主机 DRAM 获取	一次 64 字节获取
数据传输	通常半双工	全双工

5.3.3 形态因子与 NVMe-oF

- 形态因子：AIC（HHHL/FHHL 插卡，早期 PCIe 2.0/3.0）、U.2（SFF-8639，x4 lane，服务器可组合最多 48 块 U.2 NVMe）、U.3（三模 SAS/SATA/NVMe 单控制器，支持热插拔、向后兼容 U.2）、M.2（PCIe 3.0/4.0 至 x4）、EDSFF（企业数据中心标准形态因子）。

- **NVMe-oF（NVMe over Fabrics）**：将 NVMe 命令经网络传输访问远端设备，2016 年发布标准；截至 2021 年 5 月支持的传输协议：**FC（FC-NVMe）**、**TCP（NVMe/**

TCP）、Ethernet/RoCE v1/v2、InfiniBand（NVMe/IB）；软件实现包括 Linux 内核 initiator/target、SPDK、StarWind（Windows）。

- **生态支持**：Linux 3.3（2012-03）、Windows 8.1/Server 2012 R2 原生驱动、macOS Yosemite 10.10.3、iOS（iPhone 6S 起首个移动端 NVMe over PCIe）、ChromeOS、FreeBSD 10.2、VMware vSphere 6.0（VSAN 支持 NVMe）、QEMU 1.6、UEFI；Fugaku、Summit、Sierra 等超算以 NVMe 构建 burst buffer 存储层。

5.3.4 NVMe 测试与特性要点

- **SanBlaze RM5**（16 盘位 Gen5 NVMe 测试系统）：900+ 自动化测试脚本，覆盖 **ZNS**（分区命名空间）、**VDM**（厂商调试消息）、**SRIS**（独立参考时钟）、**TCG Opal** 加密、**Dual Port**（双端口）、**SR-IOV**（单根虚拟化）、**CMB**（控制器内存缓冲）/ **HMB**、**T10 DIF/DIX**（端到端数据完整性）、NVMe-MI over VDM、OCP 2.0 规范，可与协议分析仪（SerialTek）触发器联动。

- Kioxia CM7 等 Gen5 企业盘支持 SR-IOV、CMB、多流写入，双端口（dual port）高可用为企业级标配。

- **Quarch** 提供 Gen5 MCIO/U.2/U.3/E3.S 热插拔故障注入模块与 PPM 可编程电源（250 kHz 采样、4 mV 分辨率、1 µs 粒度），支持 PAM 侧带与 UNH-IOL 认证测试。

5.3.5 与晶格计算的关联点

NVMe 的多队列（65,535 队列）、无锁并行与全双工特性，是 KFNative™ **原生本地 NVMe 服务层**（Tier0 NFS enable native local NVMe for Kubernetes）实现 Tb/s 级 IO 的直接依据；KFNative 强调 **Native PCIE / 原生本地 NVMe**，区别于 NVMe-oF 经网络搬运的间接路径，并主张**消除昂贵双端口 SSD**（由跨节点纠删码替代），与"标准件、低成本"路线一致。

5.4 DDR5 技术详析

5.4.1 DDR 基本原理与命名

DDR（Double Data Rate）在时钟信号的**上升沿与下降沿均传输数据**（double pumped），同等时钟频率下数据传输率翻倍。带宽 = 传输率 × 总线宽度（64 位 = 8 字节）。注意：DDR 与"双通道（dual channel）"是两个独立概念。

命名	内存时钟	I/O 总线时钟	传输率	理论带宽（64-bit）

DDR-200 / PC-1600	100 MHz	100 MHz	200 MT/s	1.6 GB/s
DDR-400 / PC-3200	200 MHz	200 MHz	400 MT/s	3.2 GB/s
DDR3-1600 / PC3-12800	200 MHz	800 MHz	1600 MT/s	12.8 GB/s
DDR4-3200 / PC4-25600	400 MHz	1600 MHz	3200 MT/s	25.6 GB/s
DDR5-4800 / PC5-38400	300 MHz	2400 MHz	4800 MT/s	38.4 GB/s
DDR5-6400 / PC5-51200	400 MHz	3200 MHz	6400 MT/s	51.2 GB/s

5.4.2 DDR5 规格要点

项目	规格
标准传输率	PC5-4800 / PC5-5200 / PC5-5600（对应 4800 / 5200 / 5600 MT/s）
CAS 延迟	CL 40（0.416 ns @DDR5-4800）/ CL 42（0.384 ns @DDR5-5200）/ CL 46（0.357 ns @DDR5-5600）
电压	1.1 V（POD 接口）
Bank 结构	x4/x8：32 banks（8 组 × 4 bank）；x16：16 banks（4 组 × 4 bank）
通道结构	两个独立 I/O 子通道（提升带宽利用率）
边带访问	I3C-basic / I2C sideband（替代 DDR4 的 SMBus）
命令/地址	每 DIMM 两条 7-bit DDR 命令/地址总线，经 registered clock driver 转 14-bit SDR 到每颗内存芯片
拓扑	Fly-by 拓扑；终结的时钟、控制与命令/地址总线；288-pin DIMM
容错	片上 ECC、内置纠错（相较 DDR4 的显著增强）

- **LPDDR5**（JEDEC JESD209-5，2019-02-20）：I/O 速度由 LPDDR4 的 3200 MT/s 翻番至 **6400 MT/s**；64-bit 总线每秒可传 51.2 GB，128-bit 总线可破 100 GB/s；最高 16 Bank 可编程、多时钟体系；引入 Data-Copy 与 Write-X 命令降低功耗；支持链路 ECC，信号电压 250 mV，Vddq/Vdd2 仍为 1.1 V。

5.4.3 DDR5 测试技术与工具要点

- **JLA420A DDR5/LPDDR5 协议分析仪**：白皮书称其为"业界最高性能的 DDR5/LPDDR5 协议分析仪"，支持高速信号采集与总线解码；可配置 RDIMM DDR4 interposer 分析 3200 MT/s DDR4 内存条。

- **DDR5 Socket-DIMM 转接卡**：288-pin NON-ECC UDIMM（x16/x8），支持 PC5-4800/5200/5600，验证稳定运行于 DDR5 2800 MHz（PC5-5600）、低延迟时序 46-45-45 @ 1.1 V；支持多种 FBGA 封装尺寸、单 rank、I3C-basic/I2C、两个独立 I/O 子通道、fly-by 拓扑。
- **Interposer 系列**：DDR5/DDR4 DIMM interposer、LPDDR5/LPDDR4 interposer，用于把内存信号引出至示波器/逻辑分析仪/协议分析仪调试。
- **测试平台**：RDIMM/UDIMM 内存测试平台，配合 Quarch 电源/热插拔模块进行功耗与故障注入验证；LPDDR5 测试设备仍在完善中，LPDDR4X 测试需求仍存续。

5.4.4 与 KFNative™ 的关联点

DDR5 作为标准 x86 服务器主存底座，在晶格计算中承载 **KV-Cache 热点部分与计算过程临时数据**，与近端 NVMe SSD 构成"**DDR5 热层 + NVMe SSD 冷/温层**"分层缓存体系；同时 Samsung 128/512 GB CXL 内存模块以 DDR5 为基础颗粒，印证 DDR5 生态与 CXL 内存池化路线的衔接。

5.5 四类技术测试与工具综合要点

技术	关键测试工具	覆盖要点
PCIe Gen5	SerialTek KODIAK 分析仪、SerialCables Gen5 Switch 卡（PEX89000）、Retimer 卡、MCIO 线缆、JBOF	32 GT/s 线速抓包、眼图/信号完整性、链路训练与协议解码
CXL	协议感知 Retimer、一致性/互操作性测试（UNH-IOL 等）	CXL.io/cache/mem 三协议、FLIT 传输、缓存一致性语义
NVMe	SanBlaze RM5（16 盘位 Gen5、900+ 脚本）、Quarch 热插拔/PPM 电源、SerialTek 触发器联动	ZNS/VDM/SRIS/Opal/Dual Port/SR-IOV/CMB-HMB/T10 DIF-DIX、NVMe-MI over VDM、OCP 2.0、故障注入与功耗
DDR5	JLA420A 协议分析仪、DDR5 Socket-DIMM 转接卡、DDR4/5 与 LPDDR4/5 interposer、RDIMM/UDIMM 平台	PC5-4800/5200/5600 协议解码、时序/CAS、总线信号引出调试

本章结论：PCIe 5.0（32 GT/s，x16 每方向 64 GB/s）为晶格互联提供物理基座，Gen6（64 GT/s）延续翻倍演进；CXL 基于同一 PHY 提供缓存一致性内存池化，与晶格计算"资源池化 + 消除专有互连"同向互补；NVMe 以 65,535 队列与全双工并行支撑 Tb/s 级原生存储 IO；DDR5（PC5-4800~5600，1.1 V）构成主存热层。四类技术的成熟测试工具链（协议分析仪、Switch/Retimer、故障注入、interposer）证明该标准生态可直接支撑 KFNative™ Extended PCIe Fabric 的工程验证与规模化落地。

第 6 章 应用场景与性能特征

6.1 应用场景一：大规模 LLM 训练

KFNative™ Ring Extension 面向训练场景的收益主张：

- **完全绕开 HBM 和 NVLink**：训练中间数据与 KV-Cache 流式驻留近端 SSD，不再受 HBM 容量与 NVLink 带宽约束；
- **有效组织显卡级 GPU 做大规模训练**：把普通商用 GPU 组织成 Ring 拓扑，All/All Ring Reduce 保证集合通信效率；
- **大幅降低大模型训练所需的 GPU 数量**：可"以少 64 倍的 GPU 数量训练最大规模 LLM"；
- **支撑高清超长视频生成**等生成式负载。

6.2 应用场景二：LLM 推理与 KV-Cache 加速

基于第 2.4 节的流式计算模型，推理场景的典型收益：

- 模型权重常驻 SSD、逐层流式加载，推理引擎可服务超显存容量的超大模型；
- KV-Cache 以 Paged 方式就近驻留 GPU 专有缓存，数十倍提升 GPU/KV-Cache 有效带宽；
- 数据交换端到端在延伸 PCIe 内完成，延迟较网络路径有数量级改善。

6.3 应用场景三：Kubernetes 云原生存储

KFNative™提出 **Tier0 NFS enable native local NVME for Kubernetes**（面向 Kubernetes 的 Tier0 NFS 使能原生本地 NVMe），支持 Rancher、OpenShift、Amazon EKS Anywhere 等主流 Kubernetes 发行版，提供 **Tb/s 级 IO 与网络吞吐**，定位为"革命性的 Kubernetes 全闪存储方案"。

6.4 应用场景四：大数据与向量检索

晶格互联同时面向大数据分析与向量数据库：

- **KV Cache、Vector Database** 为 KFNative™ 的第一层核心承载，以原生 NVMe 吞吐支撑向量检索；
- **Boosting data analytics with native flash storage**：以原生闪存加速大数据分析；
- 支持 VR/AR/Rendering、HPC、内存池化等扩展负载。

6.5 与传统架构的性能/架构对比

竞争对手分析给出了KFNative™与传统全闪存储路线在架构层面的对比：

维度	传统全闪存阵列（AFA）路线	KFNative™路线
内部总线	NVMe over Fabric（经网络搬运，C-node/D-node/ Switch 架构）	Native PCIE / Extended PCIe（对称优化架构）
可靠性	依赖双端口 SSD + 跨 D-node 纠删码	消除双端口 SSD，跨对称节点纠删码（Erasure coding）
应用扩展	D-node 支持大数据与应用程序	超融合模式支持任何应用程序

在性能/成本特征上，KFNative™的优势主张：

- **Off-the-Shelf components**：标准件，无需专用硬件；
- **Utilizing server from any vendor**：任意厂商服务器均可利用；
- **High performance and low CPU utilization**：高性能且低 CPU 占用；
- **Native local NVME performance**：原生本地 NVMe 性能；
- **Power efficient**：高能效；
- **Converged SSD/Memory Caching**：SSD/内存融合缓存；
- **Native NVME throughput for KV cache**：为 KV-Cache 提供原生 NVMe 吞吐。

6.6 性能数据可信度说明（事实与预测分离）

本白皮书引用的关键性能数据来源与性质如下，读者应审慎区分：

数据	性质	说明
单层加载 35 ms、计算 180 ms	设计推算示例	以 GPT-3 175B/Gen6 x16 SSD 100 GB/s 给出的估算，非实测
64X fewer GPUs	产品主张/目标	未见公开实测数据佐证
数十倍有效带宽提升、微秒级延迟	架构设计推论	基于路径缩短与拓扑并行的定性/定量估计

千亿/近千亿/数千亿市场规模	行业量级引用	市场定位引用，未注明具体研究机构出处

本章结论：KFNative™在推理流式计算、大规模训练成本、Kubernetes 存储三个场景具有结构性优势（原生 PCIe 通路、消除双端口 SSD、标准件低成本）；但 35ms/180ms、64X 等关键数据属于设计推算或产品主张，尚未形成公开实测验证，评估时需结合实测进展审慎引用。

第 7 章 市场策略与总结

7.1 市场空间与竞争格局

KFNative™ 给出的市场定位：

市场	规模	KFNative™ 优势主张
全闪存阵列（AFA）	约千亿（Major Player：Pure、NetApp 等）	Off-the-Shelf 组件、任意厂商服务器、高性能低 CPU 占用
超融合（HCI）	近千亿	原生本地 NVMe 性能、高能效、面向 Kubernetes 的革命性全闪方案
数据中心/算力	数千亿	SSD/内存融合缓存、KV-Cache 原生 NVMe 吞吐、原生闪存加速大数据分析

竞争路线上，KFNative™与传统全闪存阵列的关键差异在于：传统方案以 **NVMe over Fabric 作为内部总线**（C-node/D-node/Switch 架构），依赖双端口 SSD 与跨 D-node 纠删

码；KFNative™以 **Extended PCIe 作为内部总线**（对称优化架构），消除双端口 SSD、跨对称节点纠删码，并通过超融合模式支持任何应用程序。

7.2 市场策略

KFNative™ 市场策略按地域与路径展开：

- **国内**：与一线 SSD 厂商合作主打企业级市场；与系统集成商合作增加产品附加价值与可区分性，进入私有云市场；以中小企业、学区为出发点逐步推广。
- **海外**：找品牌服务器厂商合作进入企业级市场；与一线 GPU 厂商合作进入数据中心；与系统集成商合作拓展企业级市场；以大学和研究机构为出发点逐步推广。

7.3 技术定位与差异化总结

综合全篇，KFNative™的技术定位可概括为：

12. **技术内核**：KFNative™（KV Cache、Vector Database）——GPU/SSD 配对引擎与流式计算核心；
13. **互联基座**：Lattice PCIE Fabric——延伸 PCIe 的统一数据平面，实现 GPU pooling 与 NVMe SSD pooling；
14. **存储服务**：Tier0 NFS / Native NVMe——面向 VM/Kubernetes 的云原生高性能存储；
15. **差异化**：与 Traditional Server/HCI/AFA/AFA NVMe-oF 等既有路线相比，KFNative™以"标准件 + 紧耦合 + 原生 PCIe 通路"实现低成本、低延迟、高能效与任意厂商硬件兼容。

7.5 结论与展望

核心结论：晶格计算通过 KFNative™ 将 GPU 与 SSD 以 Extended PCIe Fabric 紧耦合，把 KV-Cache 与模型权重流式化到近端 NVMe 存储，从根本上绕开 HBM 容量与 NVLink 互连约束，实现延迟从毫秒级到微秒级的数量级改善与数十倍有效带宽放大；该路线与 PCIe/CXL 互连演进、NVMe 多队列并行、DDR5 内存生态天然协同，有望在千亿级全闪存储与超融合市场形成差异化竞争力。

展望与风险提示：KFNative™的关键性能主张（如 64X fewer GPUs、流式计算 35ms/180ms）目前主要基于架构设计与推算，尚需通过产品化实测验证；其竞争力高度依赖 PCIe Switch/Gen6 SSD 等标准组件的代际演进节奏，以及与传统 AFA 巨头、CXL 生态路线之间的市场博弈。KFNative™端到端实现的性能与成本目标，其"绕开 HBM/NVLink"的开

放路线将对现有 AI 算力成本结构产生实质影响。