Tensorfer CNCC2025

GD2FS：面向AI的新一代分布式文件系统

皮振伟 <pizhenwei@tensorfer.com>

2025-10

张量跃迁

AI时代业务负载的新特征

AI场景下，计算机求解问题的三要素的变化

- 资源
  - 计算：大量的运算由GPU完成
  - 存储：更多、更快的NVMe设备；GPFS的大量使用
  - 网络：200Gbps、400Gbps网络大规模使用；RDMA大规模使用
- 数据
  - 数据由KB、MB级变成GB、TB级
- 程序
  - 各类的逻辑运算变成了模型的数据运算

AI场景下，资源的变化

处理器	核心数	FP64 FLOPS	FP32 FLOPS	AI 算力
AMD EPYC 9654	96	~1.42 TFLOPS	~2.84 TFLOPS	N/A
NVIDIA H100 SXM	132 * 64 * 32	~34 TFLOPS	~67 TFLOPS	~3958 TFLOPS (FP8)

常见的CPU服务器使用2 NUMA nodes, GPU服务器使用8 GPUs，单台服务器的计算性能相差多个数量级。

数据来源Deepseek

- 传统的逻辑计算
  - 规则驱动，基于知识推理：将人类知识编码成明确的逻辑规则

- AI的数据计算
  - 数据驱动，从样例中学习：让机器从大量数据中自动发现模式和规律

因此，AI场景相比于逻辑计算，数据变得越来越多，程序变得越来越少。以及，资源的消耗越来越多。

Tensorfer

AI驱动的存储需求变革：现状与挑战分析

- 训练
  - 大文件场景，保存Checkpoint，以及从checkpoint恢复训练
  - 大量小文件场景，GPFS大量应用

- 推理
  - KV Cache卸载。KV Cache是否命中，成本相差约**10**倍
  - 引擎冷启动，模型加载，例如DeepSeek-R1-0528约**642GB**
    - 云计算Spot实例
    - 研发效能

- 训推一体
  - 保存训练结果、推理调试

- KV Cache卸载
  - vLLM/sglang均使用Host memory缓存KV Cache。目前命中率普遍不高
  - 分布式存储的KV Cache能够提升KV cache命中率
  - TTFT(Time To First Token)是LLM服务的核心指标，存储延迟直接决定产品体验
  - KV Cache可被重计算，原则上可以丢失

- 推理引擎冷启动
  - 提升Spot实例的有效服务时间
  - 提升研发效能
  - 进一步地，是否可以使用K8S调度推理引擎实例，甚至FaaS化?

GD2FS的核心设计

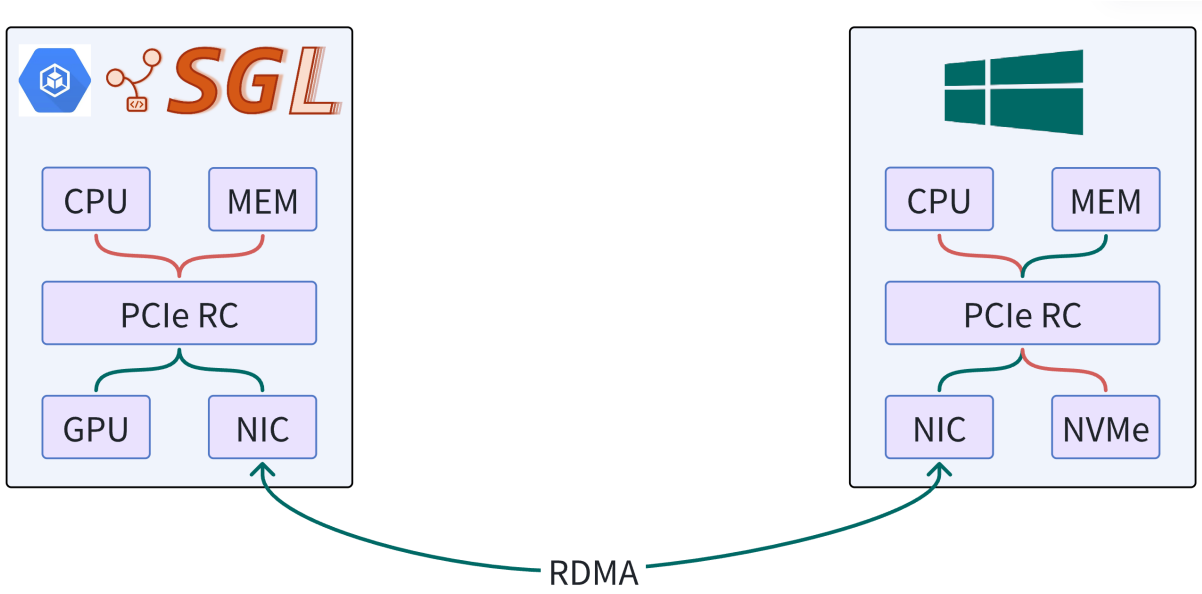
关键存储设备的性能分析

	GPU	DDR5	网卡	NVMe
吞吐	2TB/s * 8 = 16TB/s	8 channels ~300GB/s	400Gbps * 4 = 200GB/s	W 2GB/s * 4 = 8GB/s R 6GB/s * 4 = 24 GB/s
延迟	~100+ ns	~80+ ns	1K RDMA ~2us 1K TCP ~6us	4K Write ~20us 4K Read ~80us

AI场景下的主要运算发生在GPU，数据的读取、写入过程中，NVMe是性能瓶颈。

GD2FS是**GPU Direct Distributed File System**的缩写，其命名即设计理念：

- 深度融合 GPU 加速与高速网络能力，支持GPU Direct RDMA
- 同时支持TCP多网卡、多流并发



分布式高性能缓存的优势

- Linux Kernel Page Cache
  - 单机作用域，不支持分布式共享
  - Page(4KB)级别的回收
  - kswapd(per NUMA)的性能瓶颈

- GD2FS的分布式缓存
  - 分布式作用域，全局共享
  - 文件级别的回收
  - 多核更加友好

- 用户态内存管理
  - Page Size：内核4K粒度过小，采用更大的"Page Size"，提升传输效率
  - 内存淘汰策略：整个文件级别淘汰，避免系统颠簸
  - 大页：支持Hugetlbfs的匿名映射和文件映射
  - Zero Copy：数据无额外的复制

- 用户自定义缓存策略
  - 支持分布式FADV-WILLNEED，预读数据到缓存
  - 支持分布式FADV-DONTNEED，精确释放缓存

弹性副本策略

- 单副本
  - 契合KV Cache可重计算特点
  - 更低的成本、更大的存储容量
  - 更低的延迟，write back方式极速写入
- 多副本
  - 持久化：Checkpoint、模型文件使用经典的三副本，防止数据丢失
  - 缓存：1-N副本弹性伸缩，防止推理引擎的启动风暴

全异步、Scatter/Gather API

- 提交I/O

```
ssize_t xfer_gd2fs_preadv(xfer_gd2fs_ctx *ctx, void *id, const char *filepath, uint64_t off,
const xfer_gd2fs_sge *sges, uint16_t nsge, uint16_t flags);
```

- 等待完成

```
typedef struct xfer_gd2fs_completion {
   void *id; /* submission passed back */
   uint64_t value;
   uint16_t status;
   uint8_t reserved[6];
} xfer_gd2fs_completion;

int xfer_gd2fs_wait(xfer_gd2fs_ctx *ctx, xfer_gd2fs_completion *completions, int
maxcomp, int timeout_ms);
```

Python支持

- 提交I/O

```
def Read(self, filepath: str, offset: int, sges: list[SGE], flags: int, desc: str) -> Request:
    """
    Returns: Request object for tracking operation
    """
```

- 等待完成

```
def Wait(self, timeout: int) -> list[Request]:
    """
    Returns: List of completed requests
    """
```

GD2FS的典型性能指标

FIO性能指标

RDMA	RW	64M (AVG us)	256M (AVG us)	1G (AVG us)
GPU - GD2FS	读	1529	5938	23601
	写	4400	9975	35774

TCP	RW	64M (AVG us)	256M (AVG us)	1G (AVG us)
DDR - GD2FS	读	4249	16673	67280
	写	10654	36757	123187

INTEL(R) PLATINUM 8563C; NVIDIA H20 96GB; Mellanox CX7 400Gbps; INTEL SSDPF2KX038T1

模型下载

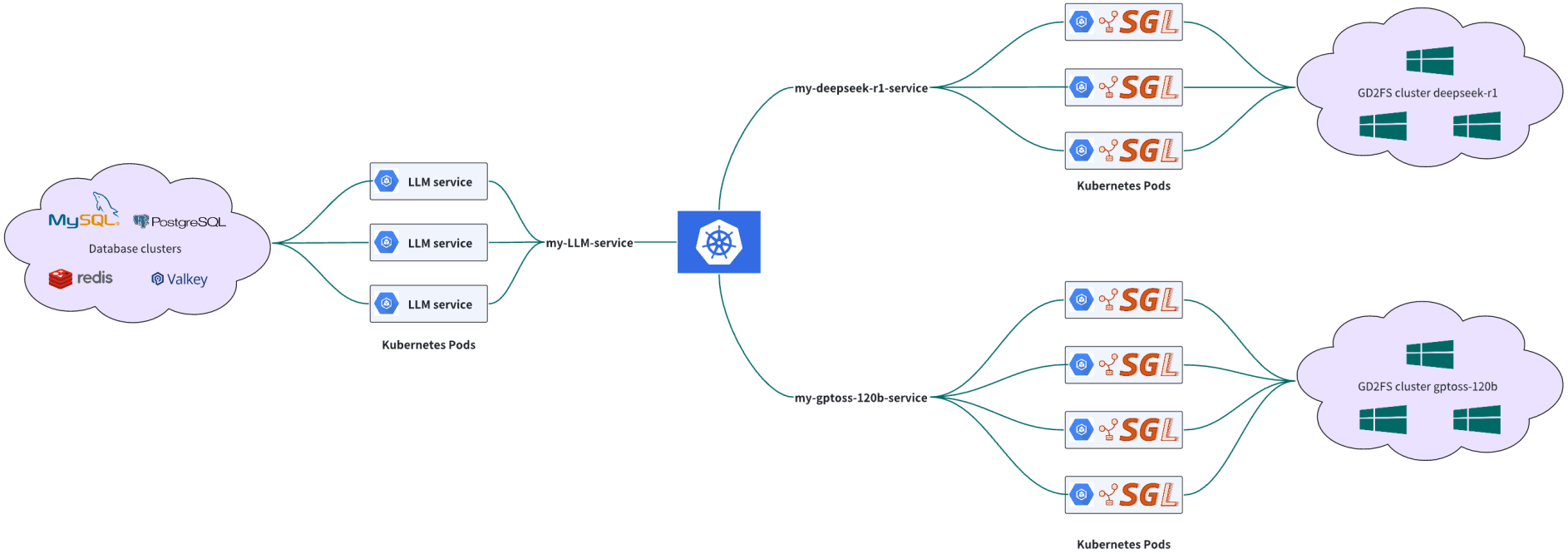
TCP	模型	Kimi-K2-Instruct	DeepSeek-R1-0528	Qwen3-Omni-30B-A3B-Instruct
	模型大小	959G	642G	66G
400G * 1	GD2FS Cache Miss（NVMe * 1）	154 +/- 3 s	105 +/- 2 s	11 +/- 1 s
	GD2FS Cache Miss（NVMe * 2）	88 +/- 3 s	57 +/- 2 s	7 +/- 1 s
	GD2FS Cache Hit	37 +/- 2 s	25 +/- 1 s	3 +/- 0.2 s
400G * 2	GD2FS Cache Hit	32 +/- 2 s	19 +/- 1 s	2.5 +/- 0.2 s

INTEL(R) PLATINUM 8563C; NVIDIA H20 96GB; Mellanox CX7 400Gbps; INTEL SSDPF2KX038T1

推理架构演进展望

Tensorfer

Toward stateless LLM inference accross GPU clusters



开发、验证中，预计2026 Q1发布数据、文档和相关论文

张量跃迁

Thank You