sglang 仓库 2026 年第 26 周周报（06-22 至 06-28）

sgl-project/sglang

周期：2026-06-22 至 2026-06-28

来源 PR：288・重点 PR：24・自动生成

原文链接：http://prhub.com.cn/sgl-project/sglang/reports/2026-06-22-to-2026-06-28

执行摘要

本周代码变更活跃，共处理 288 个 PR（24 个重点 PR），平均重要性 6.2。变化集中在推测解码优化、分布式缓存扩展、新模型集成以及基础设施建设。DeepSeek/GLM 系列的 kernel 融合与 JIT 迁移持续深化，扩散模型生态新增 JoyEcho 和 Krea-2 等重要模型。同时，IPC 序列化升级和配置注解化重构提升了长期可维护性。风险方面，“缺少测试覆盖”和“核心路径变更”依旧突出，建议后续加强测试配套。

本周重点变化

- 推测解码性能提升：Kimi-K2.5 Eagle3 的 draft fc 分片与对称内存 all-gather 技术节省 70-80μs *decode* 时间；DSA 索引器 Q/K 路径融合实现 ~10% *decode* 加速；ReplaySSM 缓冲解码使线性注意力 HBM 流量减半。
- MiniMax-M3 支持推进：作为大型拆分的一部分，本周合并了 Split 1/4（稀疏注意力操作 /JIT kernel）和 Split 2/4（mem-cache / HiCache / sparse KV pool），为后续组装奠定基础。
- 扩散模型生态扩张：新增 JoyEcho 多镜头音视频生成（基于 LTX-2 骨干 + 内存银行机制）和 Krea-2 单流 MMDiT 文生图模型，同时优化了 LTX2.3 和 Qwen-Image 的预处理与 kernel 性能。
- 基础设施重构成熟：IPC 数据类从 dataclass 迁移至 msgspec.Struct（#28688），序列化效率提升并为后续支持 msgpack 做准备；ServerArgs 全面注解化（#28919），减少约 2400 行 add_cli_args 代码。
- 分布式与缓存能力增强：DeepSeek-V2 支持 Decode Context Parallel（#14194），扩展上下文长度；Session Radix Cache（#27058）替代固定槽位，避免高并发下 worker wedge；NIXL FILE 后端添加 L3 缓存清理器（#28258）。

模块与主题趋势

推测解码（Speculative Decoding）

本周推测解码相关 PR 数量密集，涵盖 Eagle3、DSA、DFLASH 等多个方案。核心趋势是：（1）kernel 融合减少 kernel launch 与 HBM 访问；（2）CUDA Graph 捕获范围扩大，如将 draft 采样纳入 graph；（3）统一不同解码路径的结果处理与 relay 通信。代表 PR 包括 #29223、#27705、#28492 等。

DeepSeek / GLM 模型优化

继续推进 JIT 迁移工作：dsv3_router_gemm 完全从 AOT 迁移（#21531），fused A GEMM 新增 CuTe DSL 实现（#27397）。BCG（Breakable CUDA Graph）取代 PCG 用于 GLM5（#27053），在 B200 TP8 上带来 ~4% 吞吐提升和 10% TTFT 降低。双流 MoE 的 routed experts 执行流交换（#29142）后被回退（#29452），表明此方向尚需评估。

扩散模型

本周扩散模型迎来多个重要PR：JoyEcho多镜头生成（#27420）、Krea-2支持（#29052）、LTX2.3 优化（#28624、#29390）、Qwen-Image NVFP4 量化（#28928）等。模型生态快速扩展，同时围绕 cache-dit、causal attention cache 和 warmup 等基础设施也在完善。

基础设施重构

IPC 序列化：msgspec.Struct 迁移（#28688）和 SamplingParams 转换（#29198）是第一步，后续预期逐步移除 pickle。ServerArgs 注解化（#28919）显著减少样板代码。benchmark 工具移动至 sglang.benchmark 子包，统一入口。

多硬件平台

AMD 平台活跃度维持高位：新增 MI355X 分解式夜间测试、DSV4 decode reduce_scatter 优化（#29103）、BCG 启用（#27833）、MXFP8 Triton kernel（#28211）等。NPU 与 Intel XPU 也有若干 bugfix 和文档更新。

风险观察

1. 缺少测试覆盖（40 次标注）：新模型与优化路径普遍缺乏配套测试，如 MiniMax-M3 的 Split 合并后尚待精度测试，ReplaySSM、DCP 等新功能测试不足。
2. 核心路径变更（39 次标注）：调度器、模型初始化、IPC 等核心模块的修改可能引入回归，特别是 IPC 序列化过渡期需确保 wrap/unwrap 成对。
3. 性能回归风险：部分优化依赖特定硬件（Blackwell、SM120）或编译环境（JIT 与 AOT 对齐），其他平台可能出现性能差异。BCG 与 PCG 的切换存在条件判断风险。
4. 平台兼容性：MUSA 平台对 JIT 支持有限，可能仍需保留 AOT 版本；AMD HIP 路径的精度问题（如 fp8 KV path）仍在修复中。

重点 PR 速览

- #29223：Kimi-K2.5 Eagle3 draft fc 分片 + 对称内存 AG— 通过分片 draft fc 权重并使用基于 PyTorch 对称内存的 Triton kernel 替换两个 NCCL all-gather，decode 步骤节省 70-80μs。
- #27705：DSA 索引器 Q/K 路径融合— 将 key projection 与 head-gate projection 融合为单一 matmul，Q/K 路径各融合为单 kernel，decode 性能提升约 10%。
- #27397：JIT fused A GEMM 支持 GLM-5 和 SM120— 添加 CuTe DSL 和 CUDA JIT kernel 实现，在 SM120 上自动选用，性能提升约 2%，并为未来移除 AOT 版本埋下伏笔。
- #28713 & #28712：MiniMax-M3 Split 1/4 & 2/4— 分离式引入稀疏注意力操作、JIT kernel 和 KV 缓存池，是 MiniMax-M3 完整支持的重要里程碑。

- #29186：Baidu Unlimited-OCR 模型集成— 复用 DeepSeek OCR 视觉编码器，新增预填充感知 SWA 和纯 SWA radix 缓存，展示自定义注意力模型集成范例。
- #28688：IPC 数据类迁移至 msgspec.Struct— 涉及 25 个文件，引入 PickleWrapper 渐进迁移，是 IPC 序列化升级的第一步。
- #14194：DeepSeek-V2 Decode Context Parallel— 通过 KV 缓存虚拟化与注意力输出校正实现上下文并行，使 8×H20 上支持更长上下文。
- #27058：Session Radix Cache— 用 radix 缓存标签替代固定槽位，避免高并发下 worker wedge，默认关闭但开启后可显著提升 session 场景效率。

后续建议

1. 加强测试覆盖：针对本周引入的新模型（MiniMax-M3、Baidu Unlimited-OCR、Krea-2）、新缓存策略（ReplaySSM、Session Radix Cache）以及分布式功能（DCP），建议尽快补全精度测试和端到端 CI。
2. 推进 IPC 过渡：在 #28688 基础上，逐步移除 PickleWrapper 和 pickle 默认值，统一使用 msgspec msgpack。需确保所有传输路径正确调用 wrap/unwrap。
3. 关注性能回归：Fused A GEMM 的 AOT 版本移除后，需持续监控其他平台（MUSA）下的性能；BCG 替代 PCG 后需在更多模型和硬件上验证。
4. 文档与示例同步：随着多项新功能和模型加入，文档和 cookbook 需及时更新，特别是部署参数和硬件配置。