# PR #23836 完整报告

sgl-project/sglang

[diffusion] chore: change default seed to 42

合并时间：2026-04-28 20:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23836

## 执行摘要

- 一句话：默认种子改为 42，重构 partitioning 与 disagg 种子传递
- 推荐动作：建议关注 partitioning 模块提取的设计（单一职责、可复用）、DenoisingContext 数据类与字典返回的权衡、disagg 种子传递的演进。该 PR 展示了增量重构与特性增强的合并模式，适合用于学习渐进式重构。

## 功能与动机

来自 PR body：'to align with diffusers default: generator = torch.Generator().manual_seed(42)'。使默认种子与 diffusers 库一致，保证复现性。

## 实现拆解

1. 修改默认种子值：在 http_server.py 中将 DEFAULT_SEED 常量从 1024 改为 42；在 image_api.py、video_api.py 中调整 Form(...) 默认参数；在 protocol.py 中更新 ImageGenerationsRequest 和 VideoGenerationsRequest 的 seed 字段默认值；在 io_struct.py 中更新 RolloutRequest 的 seed 默认值。
2. 抽取通用 Partitioning 模块：创建 python/sglang/multimodal_gen/test/partitioning.py，定义 PartitionItem 数据类和 partition_items_by_lpt 函数实现 LPT（最长处理时间）分区算法。随后在 run_suite.py 中导入该模块重写 auto_partition 函数，并新增 get_suite_files_rel 和 partition_items_by_index；在 CI 脚本 compute_diffusion_partitions.py 中通过 _load_partitioning_helpers 动态加载该模块，移除本地重复实现，增加对 b200 suite 的支持和 include_standalone 选项。
3. 引入 DenoisingContext 数据类：在 denoising.py 中新增 DenoisingContext dataclass，替代先前 _prepare_denoising_loop 返回的字典。在 hunyuan3d_shape.py 中改为返回 DenoisingContext 实例，并新增 is_warmup 字段。
4. 增强 Disagg 种子传递：在 scheduler_mixin.py 的 extract_transfer_fields 中添加 seed 序列化逻辑，将 req.seed 写入 scalar_fields；在 _build_disagg_req 中根据 seed 字段重建 torch.Generator，并支持 seed 为列表的多输出场景。
5. 补充测试覆盖：在 test_disagg_trace.py 中新增两个测试用例验证单种子和列表种子在 disagg 中的正确传递与重建。
6. 更新 CI 配置：修改 diffusion-ci-gt-gen.yml 以支持 b200 suite 和新 partitioner，同步更新 gen_diffusion_ci_outputs.py。

关键文件：

- python/sglang/multimodal_gen/test/partitioning.py（模块 测试分组；类别 test；类型 test-coverage；符号 PartitionItem, partition_items_by_lpt）：新增文件，提取了 LPT 分区算法的核心实现（PartitionItem 数据类和 partition_items_by_lpt 函数），被测试运行器和 CI 脚本共同引用，是本次重构的关键产出。
- python/sglang/multimodal_gen/runtime/disaggregation/scheduler_mixin.py（模块 反聚合调度；类别 source；类型 core-logic；符号 extract_transfer_fields, _build_disagg_req）：核心逻辑变更：在提取传输字段时序列化 seed，在反序列化时支持种子列表重建多个 generator，使 disagg 正确传递种子信息，支撑多输出场景。
- scripts/ci/utils/diffusion/compute_diffusion_partitions.py（模块 CI 脚本；类别 infra；类型 infrastructure；符号 _load_partitioning_helpers, build_partition_items）：CI 分区脚本重构：改为从 partitioning.py 导入 PartitionItem 和 lpt_partition，移除重复实现；增加对 b200 suite 的支持和 standalone 可选项。
- python/sglang/multimodal_gen/test/run_suite.py（模块 测试运行器；类别 test；类型 test-coverage；符号 get_suite_files_rel, partition_items_by_index, partition_test_files）：测试运行器重构：导入 partitioning 模块替代本地 auto_partition 逻辑，新增 get_suite_files_rel 和 partition_items_by_index 函数。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_shape.py（模块 3D 形状管道；类别 source；类型 dependency-wiring；符号 DenoisingContext）：引入 DenoisingContext 数据类，替换 _prepare_denoising_loop 返回字典，新增 is_warmup 字段。
- python/sglang/multimodal_gen/test/unit/test_disagg_trace.py（模块 测试单元；类别 test；类型 test-coverage；符号 test_transfer_keeps_seed_needed_to_rebuild_generator, test_build_disagg_req_rebuilds_generator_list）：新增两个测试用例，覆盖种子在 disagg 中的正确传递和列表重建。

关键符号：PartitionItem, partition_items_by_lpt, _load_partitioning_helpers, build_partition_items, get_suite_files_rel, partition_items_by_index, extract_transfer_fields, _build_disagg_req, test_transfer_keeps_seed_needed_to_rebuild_generator, test_build_disagg_req_rebuilds_generator_list

## 关键源码片段

### python/sglang/multimodal_gen/test/partitioning.py

新增文件，提取了 LPT 分区算法的核心实现（PartitionItem 数据类和 partition_items_by_lpt 函数），被测试运行器和 CI 脚本共同引用，是本次重构的关键产出。

```
from __future__ import annotations

from dataclasses import dataclass


@dataclass(frozen=True)
class PartitionItem:
```

```
    """单个可分区项，包含种类、唯一标识和预估耗时。"""
    kind: str # 'case' 或 'standalone'
    item_id: str
    est_time: float
    used_fallback_estimate: bool = False


def partition_items_by_lpt(
    items: list[PartitionItem], num_partitions: int
) -> list[list[PartitionItem]]:
    """
    LPT（最长处理时间）分区算法。

    将 items 按预估耗时降序排列，然后依次分配给当前总耗时最小的分区，
    使得各分区总耗时尽可能均衡。
    返回包含 num_partitions 个分区的二维列表。
    """
    if not items or num_partitions <= 0:
        return []

    # 按预估耗时降序排列；耗时相同则按 kind 和 item_id 排序保证确定性
    sorted_items = sorted(
        items,
        key=lambda item: (-item.est_time, item.kind, item.item_id),
    )
    partitions: list[list[PartitionItem]] = [[] for _ in range(num_partitions)]
    partition_sums = [0.0] * num_partitions

    for item in sorted_items:
        # 找到当前总耗时最小的分区
        min_idx = partition_sums.index(min(partition_sums))
        partitions[min_idx].append(item)
        partition_sums[min_idx] += item.est_time

    return partitions
```

# 评论区精华

Review 中 gemini-code-assist[bot] 建议将各入口的默认种子改为 None 而非 42，理由是避免与 SamplingParams 默认值重复，降低维护成本。但作者未采纳，保留了显式 42，可能认为显式默认更清晰。该建议在 http_server.py、image_api.py、video_api.py、protocol.py、io_struct.py 等多处提出，均未被采纳。

- 默认种子应改为 None 还是硬编码 42 (design): 作者未采纳，保留了显式 42。可能认为显式默认更清晰或希望立即对齐 diffusers。

# 风险与影响

- 风险：默认种子从 1024 改为 42 会使原有依赖特定种子的生成结果变化，用户需适配；硬编码 seed 在各入口可能随 SamplingParams 默认值变化而不一致；partitioning 重构可能影响 CI 分片均衡，需关注后续执行情况；disagg 种子列表支持是新增路径，在极端边 case 下可能存在未覆盖的时序问题。
- 影响：用户：API 默认种子变化，生成结果与之前不同，需关注复现性。系统：disagg 服务现在支持多输出种子列表，提高功能完整性，但新增逻辑需要监控。团队：partitioning 模块化降低了后续维护成本，DenoisingContext 统一了参数传递模式。影响范围限于 multimodal_gen 子系统。
- 风险标记：默认种子变更，硬编码 seed 冗余，CI 分区重构，disagg 种子列表支持

## 关联脉络

- 暂无明显关联 PR