PR #51726 完整报告

vllm-project/vllm

[Config] Update default `_max_num_batched_tokens` from 8192 to 16384

合并时间：2026-08-11 23:09

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51726

执行摘要

- 一句话：大显存设备默认批处理 token 上限提升至 16384
- 推荐动作：值得快速浏览，改动量小但语义明确。核心看两点：一是按显存分档的默认值策略（高显存档位放开在线 server 的 token 预算），二是作者删除 " 镜像测试 " 的取舍——这类测试维护成本高且校验价值低，删除是合理决策，但建议后续为分档逻辑补充参数化测试。若你负责部署配置或调度性能调优，建议配合阅读 PR#51725 了解性能收益的前提条件。

功能与动机

PR body 说明："Update this num to a larger num when gpu memory is enough, perf can be seen PR#51725. Note that SGLang use the same num"，即当显存充足时应放大 max_num_batched_tokens 以获得更高吞吐；为便于 review 与问题回滚，该默认值变更与 PR#51725 拆分为两个 PR。代码中保留了 Kuntai 的历史注释：A100 上过大值会降低吞吐（PR#17885），因此对 A100 不使用新默认值。

实现拆解

1. 变更入口：vllm/engine/arg_utils.py 中的 EngineArgs.get_batch_defaults 类方法，这是 engine 端调度默认参数的统一出口，按 UsageContext（离线 LLM_CLASS / 在线 OPENAI_API_SERVER）返回 token 与 seq 默认值，最终写入 SchedulerConfig。
2. 分档逻辑调整：原有 ">= 70 GiB 且非 A100" 单分支拆成两级——新增 device_memory >= 160 * GiB_bytes 档（目标 B200/B300），OPENAI_API_SERVER 的默认 token 上限从 8192 提升到 16384，LLM_CLASS 保持 16384；70-160 GiB 档（H100/H200/MI300x）行为不变；其余设备（含 A100）维持 8192/2048。A100 排除与 Kuntai 历史注释保留，避免重蹈 PR#17885 的吞吐回退。
3. TPU 特例：TPU 芯片级覆盖逻辑（V6E/V5E）位于分档之后，本 PR 未改动，TPU 行为保持原样。
4. 测试配套：tests/v1/engine/test_engine_args.py 删除 test_defaults_with_usage_context（约 29 行）并移除 VllmConfig 导入。作者观点是该测试只是 get_batch_defaults 实现的逐行复制，当默认值变化时必须同步改写，无法独立暴露回归，故无保留价值；删除后未新增针对 160 GiB 新分支的测试。

5. 提交与 CI：共 2 个 commit，第二个修复 pre-commit 签名问题；合并前已触发 Buildkite CI #83202，无其他配置或文档配套改动。

关键文件：

- vllm/engine/arg_utils.py（模块 参数配置；类别 source；类型 core-logic；符号 EngineArgs.get_batch_defaults）：核心变更文件：EngineArgs.get_batch_defaults 默认值分档从两档调整为三档，新增显存 >= 160 GiB（B200/B300）档位，在线服务默认 max_num_batched_tokens 提升至 16384，并相应调整注释描述。
- tests/v1/engine/test_engine_args.py（模块 配置测试；类别 test；类型 test-coverage；符号 test_defaults_with_usage_context）：配套测试文件：删除 test_defaults_with_usage_context 及 VllmConfig 导入，体现作者对 " 镜像测试 " 维护成本的取舍。

关键符号：EngineArgs.get_batch_defaults, test_defaults_with_usage_context

关键源码片段

vllm/engine/arg_utils.py

核心变更文件：EngineArgs.get_batch_defaults 默认值分档从两档调整为三档，新增显存 >= 160 GiB（B200/B300）档位，在线服务默认 max_num_batched_tokens 提升至 16384，并相应调整注释描述。

```
    @classmethod
    def get_batch_defaults(
        cls,
        world_size: int,
    ) -> tuple[dict[UsageContext | None, int], dict[UsageContext | None, int]]:
        from vllm.usage.usage_lib import UsageContext

        default_max_num_batched_tokens: dict[UsageContext | None, int]
        default_max_num_seqs: dict[UsageContext | None, int]

        # 当用户未显式覆盖时，按 UsageContext（离线 LLM_CLASS / 在线
        # OPENAI_API_SERVER）与硬件显存规模，分别给出 batch token 与
        # seq 数量的默认值。显存探测失败时按最保守档位兜底。
        try:
            device_memory = current_platform.get_device_total_memory()
            device_name = current_platform.get_device_name().lower()
        except Exception:
            # 跨平台导入（如 Ray 场景下 import vLLM 的机器无 GPU）时
            # 无法取得设备信息，使用保守默认值。
            device_memory = 0
            device_name = ""

        # NOTE(Kuntai): A100 上设置过大的 max_num_batched_tokens 会降低
        # 吞吐，见 PR #17885，因此 A100 走最后的保守档位。
        if device_memory >= 160 * GiB_bytes:
            # 新增档位：B200/B300 等 >= 160 GB 显存设备，离线与在线
```

```
        # 服务默认 batch token 上限统一提升到 16384，配合
        # PR #51725 自适应调度预算后可获得更高吞吐。
        default_max_num_batched_tokens = {
            UsageContext.LLM_CLASS: 16384,
            UsageContext.OPENAI_API_SERVER: 16384,
        }
        default_max_num_seqs = {
            UsageContext.LLM_CLASS: 1024,
            UsageContext.OPENAI_API_SERVER: 1024,
        }
    elif device_memory >= 70 * GiB_bytes and "a100" not in device_name:
        # 原档位：H100/H200/MI300x 等大显存非 A100 设备，
        # 离线默认 16384，在线服务仍保持 8192。
        default_max_num_batched_tokens = {
            UsageContext.LLM_CLASS: 16384,
            UsageContext.OPENAI_API_SERVER: 8192,
        }
        default_max_num_seqs = {
            UsageContext.LLM_CLASS: 1024,
            UsageContext.OPENAI_API_SERVER: 1024,
        }
    else:
        # 其余硬件（含 A100）：保持原有保守默认值。
        default_max_num_batched_tokens = {
            UsageContext.LLM_CLASS: 8192,
            UsageContext.OPENAI_API_SERVER: 2048,
        }
        default_max_num_seqs = {
            UsageContext.LLM_CLASS: 256,
            UsageContext.OPENAI_API_SERVER: 256,
        }

    # TPU 场景会在此分档基础上再覆盖为芯片级默认值（V6E/V5E 等分支），
    # 该段逻辑本 PR 未改动，仅在分档完成后追加执行。
    if current_platform.is_tpu():
        chip_name = current_platform.get_device_name()
        # ... 各 TPU 芯片特例分支与原实现保持一致 ...
```

评论区精华

核心讨论集中在测试删除决策上。作者 yewentao256 在 tests/v1/engine/test_engine_args.py 的 diff hunk 上留言："For reviewers, I delete this test as it is just a copy of current code, doesn't make sense"，即 test_defaults_with_usage_context 逐行复刻 get_batch_defaults 的实现，属于 " 镜像测试 "，当实现变更时测试必须同步修改，无法独立暴露回归，因此删除。该说法未收到反对，PR 由 mgoin 直接批准合并。除此之外无其他人工讨论；claude[bot] 的评论为仓库预配置的模板提示，不构成实质 review。

- 删除 test_defaults_with_usage_context 的理由 (testing): 无反对意见，PR 获得 mgoin 批准；测试随本 PR 删除，未补充替代测试。

风险与影响

- 风险：
  1. 默认值全局行为变化：未显式设置 --max-num-batched-tokens 的部署会直接受影响，尤其是 >= 160 GiB 显存设备上在线服务档从 8192 翻倍到 16384，单批次 token 预算变大可能带来显存 OOM 或长尾请求排队延迟增加。
  2. A100 回归保护只覆盖设备名包含 "a100" 的卡，同样显存区间的其他特殊型号可能落入新档位，影响面虽小但值得留意。
  3. 测试覆盖下降：test_defaults_with_usage_context 被删除且未补充针对 160 GiB 新分支的测试，后续默认值回归主要依赖人工验证与 CI。
  4. 该参数同时影响调度器 SchedulerConfig.max_num_batched_tokens，与 PR#51725 的预算自适应逻辑叠加后，实际调度行为需要实测确认。 - 影响：影响范围：面向所有使用默认配置的用户，尤其是 B200/B300 等 160 GiB 以上显存环境——这些环境在线与离线的默认批处理 token 上限同时提升到 16384；H100/MI300x 等 70-160 GiB 设备在线档保持 8192，无变化。影响程度中等偏上：默认配置变更属于全局性行为变化，但仅作用于大显存档位；团队需要关注新档位下的显存占用与延迟反馈，并留意是否需要对中等显存设备单独调优。 - 风险标记：默认配置全局变更，批处理 token 上限翻倍，移除测试覆盖，显存 OOM 风险

关联脉络

- PR #51725 [Perf] Adaptive budget for spec scheduled token, 55%~65% E2E TTFT Improvement: PR body 明确引用该 PR 的性能数据作为提升默认值的依据；两者属于同一调度预算优化系列，51725 提供自适应预算控制，本 PR 则放开更大 batch token 默认值以释放吞吐潜力，且刻意拆成两个独立 PR 以便回溯回滚。