PR #24250 完整报告

sgl-project/sglang

[SKILL] Upgrade sglang profile and auto_benchmark skills

合并时间：2026-05-02 10:12

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24250

执行摘要

- 一句话：升级 AI-Infra 技能包，替换 benchmark 并新增 incident triage 与分阶段 profiler 脚本
- 推荐动作：值得阅读以了解 SGLang 生态中 AI-Infra 技能包的架构和设计思路，尤其是跨框架 benchmark 配置校验和分阶段 profiler 的实现。建议后续跟进 review 中的文档改进建议。

功能与动机

开发团队需要一套维护更积极、覆盖更全的 AI-Infra 自动化技能，以支持跨框架基准测试、生产事故诊断和性能分析。原 sglang-auto-benchmark 已过时，因此被替换为 llm-serving-auto-benchmark，并额外集成了 incident-triage 和 stage-aware profiler 技能。

实现拆解

1. 删除旧技能：移除 .claude/skills/sglang-auto-benchmark/ 目录及其下的配置与参考文件。
2. 添加跨框架 benchmark 技能：引入 llm-serving-auto-benchmark/，包含 validate_cookbook_configs.py（校验 YAML 配置）、compare_benchmark_results.py（对比 JSONL 结果并排序）等脚本，支持 SGLang/vLLM/TensorRT-LLM 的统一基准测试。
3. 添加 incident triage 技能：引入 sglang-prod-incident-triage/，包含 incident_artifact_tool.py（通过 HTTP 收集服务端点和 bundle）、replay_trusted_request_dump.py（回放 pickle 格式的请求 dump 用于调试）等工具。
4. 刷新 profiler 技能：更新 llm-torch-profiler-analysis/，新增 ProbePlan dataclass、send_probe_requests 等函数支持 stage-separated workload（prefill/decode/both），并添加多个框架的启动脚本（run_sglang_torch_profile_host.sh 等）。
5. 添加 SOTA performance 技能：引入 sglang-sota-performance/（具体文件未在摘要中列出，但 PR body 提及）。
6. 调整文档路径：所有示例命令改为从仓库根目录引用 .claude/skills/...，提升可移植性。

测试与配置配套：本次不含单元测试新增，但所有新增脚本均作为独立入口并附带框架级错误处理；配置方面移除了旧的 cookbook-llm YAML，替换为新的配置文件集合。

关键文件：

- .claude/skills/sglang-prod-incident-triage/scripts/incident_artifact_tool.py（模块 技能包；类别 source；类型 core-logic；符号 request_text, request_endpoint, write_json, write_text）：新增的生产故障诊断核心脚本，通过 HTTP 收集服务端点和 bundle，支持诊断 bundle 的读 / 写 / 总结。
- .claude/skills/llm-serving-auto-benchmark/scripts/validate_cookbook_configs.py（模块 技能包；类别 source；类型 core-logic；符号 flag_name, load_yaml, _as_list, _enabled）：跨框架 benchmark 配置校验脚本，确保 YAML 配置可加载并能生成立即运行的候选命令。
- .claude/skills/llm-serving-auto-benchmark/scripts/compare_benchmark_results.py（模块 技能包；类别 source；类型 core-logic；符号 _get, _float, _bool, _mean_ttft_ms）：跨框架 benchmark 结果对比脚本，从 JSONL 中加载并排序候选结果，输出 CSV 和 Markdown。
- .claude/skills/sglang-prod-incident-triage/scripts/replay_trusted_request_dump.py（模块 技能包；类别 source；类型 core-logic；符号 normalize_mm_data_item, normalize_mm_data, normalize_request_data, to_plain_dict）：用于回放本地 pickle dump 请求的脚本，绕过 SafeUnpickler 限制，支持流式请求。
- .claude/skills/llm-torch-profiler-analysis/scripts/profile_common.py（模块 技能包；类别 source；类型 core-logic；符号 ProbePlan, unique_probe_prompt, send_probe_requests, synthetic_prompt）：profiler 公共模块，新增分阶段 workload 支持和 probe 驱动，是 profiler 技能的核心。

关键符号：request_text, request_endpoint, collect_bundle, flag_name, load_yaml, _max_required_sequence, _candidate_dicts, render_command, _get, _float, _bool, _rank_key, _is_winner_candidate, normalize_mm_data_item, normalize_request_data, to_plain_dict, request_to_json_data, load_records, run_one_request, ProbePlan, unique_probe_prompt, send_probe_requests, build_probe_plan

关键源码片段

.claude/skills/sglang-prod-incident-triage/scripts/incident_artifact_tool.py

新增的生产故障诊断核心脚本，通过 HTTP 收集服务端点和 bundle，支持诊断 bundle 的读 / 写 / 总结。

```
# 该模块提供了生产诊断 bundle 的收集和检查工具。
# 核心函数 collect_bundle 遍历预定义端点，请求并保存结果。


def request_text(
    base_url: str,
    path: str,
    token: Optional[str],
    timeout: float = 10.0,
) -> tuple[bool, int, str]:
    """向服务端点发送 GET 请求并返回文本响应。"""
    url = parse.urljoin(base_url.rstrip("/") + "/", path.lstrip("/"))
    req = request.Request(url)
    if token:
```

```
        req.add_header("Authorization", f"Bearer {token}")
    try:
        with request.urlopen(req, timeout=timeout) as resp:
            body = resp.read().decode("utf-8", errors="replace")
            return True, resp.status, body
    except error.HTTPError as e:
        body = e.read().decode("utf-8", errors="replace")
        return False, e.code, body
    except Exception as e:
        return False, -1, f"{type(e).__name__}: {e}"


def collect_bundle(
    base_url: str,
    token: Optional[str],
    outdir: Optional[str],
    timeout: float,
) -> Path:
    """收集健康检查、指标、模型信息等至本地目录。"""
    timestamp = time.strftime("%Y%m%d_%H%M%S")
    bundle_dir = Path(outdir or f"./incident_bundle_{timestamp}").resolve()
    bundle_dir.mkdir(parents=True, exist_ok=True)
    # ... 遍历 ENDPOINT_SPECS 并保存结果
    write_text(bundle_dir / "SUMMARY.txt", summary_text)
    return bundle_dir
```

评论区精华

review 主要来自 gemini-code-assist[bot]，指出文档中存在若干可维护性问题：

- 多处使用未来日期（如 2026-05-01、2026-04-22）作为占位符，可能误导用户。
- 硬编码路径（如 /data/bbuf/...），建议改用通用占位符。
- 链接固定到 TensorRT-LLM pre-release 版本（1.2.0rc6），容易失效。
- 拼写错误：TensorR-LLM 应为 TensorRT-LLM。
- 模型名称疑似笔误：qwen3_30b_a3b 应为 qwen3_32b_a3b。

所有评论均为中等优先级，未产生实质性代码改动，但给出了改进方向。开发者未在评论区回复，PR 最终被合并。

- 未来日期占位符 (documentation): 建议替换为真实过去日期或明确标注为示例；PR 合并时未修改。
- 硬编码路径 (documentation): 建议改用通用占位符如 /tmp/profiler_output；未修改。
- 框架名称拼写错误 (style): gemini-code-assist[bot] 提供了修正建议；未修改。
- 模型名称疑似笔误 (style): 建议核实并修正；未修改。

风险与影响

- 风险：低风险。这些技能属于辅助工具（.claude/skills），不涉及 SGLang 运行时核心代码，不会影响推理服务质量。主要风险在于：
  - 文档中的未来日期和占位符可能造成理解偏差。
  - 硬编码路径降低可移植性。
  - 外部仓库引用（如 TensorRT-LLM pre-release 链接）可能随时间失效。
  - 新技能与旧技能的命令接口存在路径差异，使用者需更新工作流。
  - 影响：影响范围：开发者和 SRE 人员——他们使用这些技能进行基准测试、性能分析和故障排查。影响程度：中等正面影响——获得最新的跨框架基准测试能力和 stage-aware profiler，提升自动化和诊断效率。不涉及生产环境部署或终端用户。
- 风险标记：文档日期占位符，硬编码路径影响移植，外部链接版本锁定

关联脉络

- PR #24083 Add benchmark/hicache/bench_warm_cache.py for exact warm-cache shared-prefix benchmarking: 同为 benchmark 类 PR，增加基准测试能力，与此 PR 新增的 llm-serving-auto-benchmark 技能在目标上有重叠但侧重不同（缓存预热 vs. 跨框架配置校验）。
- PR #24197 Refactor device timer, clean up metrics collector, and add fwd occupancy metric: 涉及 observability/benchmark 改进，与本 PR 的 profiler 技能模块共同推动性能分析基础设施。