GLM-5.2-w8a8 在 32×910B2 的 SGLang-Ascend 部署

1 机器配置

项	配置
节点	4 × Atlas 800I A2
NPU	每节点 8 × Ascend 910B2，64GB HBM2e（torch 可见 60.55 GiB），共 **32 卡**
NPU 互连	HCCS 全互连（节点内）；8 × 200GbE RoCE（NPU 板载口，跨节点）
主机网络	bond1（DP 协调 + gloo + HCCL OOB）
CPU / 内存	4 × Kunpeng-920（192 核）/ 2 TB DDR

2 软件 & 模型

组件	值
镜像	`lmsysorg/sglang:main-cann9.0.0-910b`（image id `1cb25cc15877`，CANN 9.0.0，driver 25.5.1）
模型	GLM-5.2-w8a8，arch=`GlmMoeDsaForCausalLM`，78 层，256 专家（每 token 选 8），1 共享专家，721G
注意力	MLA（kv_lora 512，q_lora 2048，hidden 6144）+ **DSA**（DeepSeek 稀疏注意力，index_topk 2048）
量化	msModelSlim **W8A8_DYNAMIC**（`quant_model_description.json`，181 分片，173502 动态张量）
并行	**TP32**（纯张量并行，MoE 走 TP all-reduce），上下文 **1M**（1048576），权重 24.75 GB/卡

3 为什么用 SGLang + 为什么用 TP（而非 vLLM / deepep-EP）

GLM-5.2 用 **DSA**（注意力层带 `indexer`）。`vLLM-Ascend v0.21.0rc1` 加载即报 `KeyError: indexer.wq_b.weight`（未接 DSA indexer）。只有 SGLang `main-cann9.0.0-910b` 镜像**原生注册** `GlmMoeDsaForCausalLM`。专家并行用 `--moe-a2a-backend deepep` 时，真实生成阶段跨节点 all-to-all（`internode_dispatch`）必 AICore timeout 挂死（本集群 deepep RDMA 未跑通）；改 **`--moe-a2a-backend none`（MoE 走 TP all-reduce / HCCL）**后稳定——HCCL 是本集群已验证的传输。

4 部署步骤

节点 ↔rank↔bond1：.149/0/.72（master+API）；.134/1/.33；.209/2/.50；.163/3/.38。

Step 0：每节点放 /data/models/GLM-5.2-w8a8；镜像若只有 swr.../...:main 标签，先 docker tag 成 lmsysorg/sglang:main-cann9.0.0-910b（避免触发慢 dockerhub pull 挂死）。

Step 1：每节点起持久容器：--network host --ipc host --privileged，挂 8 davinci 设备 + driver + /data/models:/models:ro + /data/logs:/sglog，sleep infinity。多机引导走 --dist-init-addr（TCP），不需 ranktable。

Step 2：每节点容器内先跑运行时补丁 patch_glm52_indexer.py（见第 6 节），再起 serve。rank0 先（建 dist-init store），隔 8s 起 rank1-3。

完整命令（容器内，<rank>/< 本机 bond1> 按上表替换，master 恒 192.168.0.72）：

```
export HCCL_IF_IP=<本机bond1>
export GLOO_SOCKET_IFNAME=bond1 TP_SOCKET_IFNAME=bond1 HCCL_SOCKET_IFNAME=bond1
export HCCL_CONNECT_TIMEOUT=7200 ASCEND_RT_VISIBLE_DEVICES=0,1,2,3,4,5,6,7
export PYTORCH_NPU_ALLOC_CONF=expandable_segments:True
export TASK_QUEUE_ENABLE=1 HCCL_OP_EXPANSION_MODE=AIV HCCL_BUFFSIZE=512
export SGLANG_SET_CPU_AFFINITY=1 USE_PA_DECODE=1 USE_PA_PREFILL=1 ASCEND_USE_FIA=1

python3 patch_glm52_indexer.py # 仅 full 层构造 DSA Indexer(shared 层无权重)

python3 -m sglang.launch_server --model-path /models/GLM-5.2-w8a8 \
  --quantization modelslim --page-size 128 \
  --nnodes 4 --node-rank <rank> --dist-init-addr 192.168.0.72:30000 \
  --tp-size 32 \
  --attention-backend ascend --device npu --dtype bfloat16 \
  --mem-fraction-static 0.72 --context-length 1048576 \
  --chunked-prefill-size 8192 --max-running-requests 64 \
  --moe-a2a-backend none \
  --cuda-graph-backend-decode full --cuda-graph-backend-prefill disabled \
  --cuda-graph-max-bs 64 \
  --disable-overlap-schedule --trust-remote-code \
  --tool-call-parser glm45 --reasoning-parser glm45 \
  --host 0.0.0.0 --port 10010 --api-key <key> --served-model-name glm52
```

5 关键参数（调到的最终值）

参数	值	说明
--quantization	modelslim	msModelSlim W8A8（非 ascend / w8a8_int8）
--tp-size	32	纯 TP（去 dp-attention，去 EP/deepep）
--moe-a2a-backend	none	MoE 走 TP all-reduce（deepep 跨节点挂死）
图模式	decode=full / prefill=disabled	= V4 的 FULL_DECODE_ONLY；eager 仅 ~2 tok/s
--cuda-graph-max-bs	64	捕获 decode 图至 bs 64
--mem-fraction-static	0.72	0.9/0.8 并发下 OOM；0.72 留激活 + 通信余量
--max-running-requests	64	限并发 batch 防 OOM；提到 64 后 RPM 翻倍
HCCL_BUFFSIZE	512	2048 时多通信域各占 2GB → HcclAllGather OOM
--context-length	1048576	1M（KV 池实测 201216 tokens / 20.59 GB/卡）

`--reasoning-parser`	glm45	thinking 默认开（`enable_thinking`=True）

6 遇到的问题与解决（按出现顺序）

问题	原因	解决
vLLM 加载 `KeyError indexer.wq_b.weight`	vLLM-Ascend 未实现 GLM DSA indexer	换 SGLang `main-cann9.0.0-910b`
w8a8 不识别	模型无 `config.quantization_config`，是 msModelSlim 格式	`--quantization modelslim`（读 `quant_model_description.json`）
加载崩 `NoneType has no create_weights`（layer3 indexer）	GLM 是 shared-indexer：仅 21 个"full" 层 [0,1,2,6,...,74] 有 indexer 权重，shared 层无 wq_b，但 `deepseek_v2.py` 无条件构造 Indexer	`patch_glm52_indexer.py`：skip_topk 计算前移，仅 full 层构造 Indexer（forward 本就以 `not skip_topk` 守卫）
warmup OOM（5.28 GiB）	`mem-fraction 0.9` 静态池过大	降 0.8
`HcclAllGather` OOM	多通信域各占 2GB HCCL buffer + KV 池挤爆	`HCCL_BUFFSIZE 512` + `mem-fraction 0.75`
deepep 真实生成挂死 `NotifyDispatchA2 aicore timeout`	跨节点 all-to-all（deepep RDMA）本集群未跑通	改 `--moe-a2a-backend none`（MoE 走 TP all-reduce）
eager 仅 ∼2 tok/s	78 层逐 op Python 派发	开图模式 `--cuda-graph-backend-decode full`（快 ∼8×）
并发压测崩 NPU OOM	高并发激活超内存	`--max-running-requests` 限 batch + `mem-fraction 0.72`
镜像拉取挂死 / 掉 SSH	`.134/.209/.163` 仅 swr 标签，`docker run` 触发慢 pull	`docker tag` 本地重命名再 run

7 性能（图模式，input 256 / output 16，容器内 localhost 压测）

并发	RPM	输出 tok/s	TPOT	TTFT p50
16	465.7	124.2	64.8 ms	1.08 s

32	862.9	230.1	68.6 ms	1.21 s
64	1339.3	357.1	80.7 ms	1.69 s
96	**1409.6**	**375.9**	106.2 ms	1.72 s

峰值 **RPM 1409.6 @ 并发 96**，0 失败。TPOT 随并发 64.8→106 ms 上升 = TP32 跨节点 all-reduce 饱和，~1410 为该 TP32 配置实际天花板。对照：eager 模式单流仅 ~2 tok/s，开 decode 图模式后单流 ~16 tok/s（~8×）。thinking 正常（响应含 `reasoning_content` 思维链）。