DeepSeek-V4-Pro-w4a8 在 32×910B2 的 vLLM-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 60.96 GiB），共 **32 卡**
NPU 互连	HCCS 全互连（节点内）；8 × 200GbE RoCE（NPU 板载口，跨节点专家 all-to-all）
主机网络	bond1 50G（2×25G，承载 DP 协调 + gloo + HCCL OOB）
CPU	4 × Kunpeng-920，192 物理核，8 NUMA
内存	2 TB DDR
存储	7 TB NVMe

2 软件版本

组件	版本
镜像	`quay.io/ascend/vllm-ascend:v0.21.0rc1`
vLLM / vLLM-Ascend	0.21.0 / v0.21.0rc1
torch / torch_npu	2.10.0
Python	3.12.13
CANN / 驱动	9.0.0 / 25.5.1
模型	DeepSeek-V4-Pro-w4a8-mtp（ModelScope，放每节点 `/data/models/`）

3 部署步骤

节点 ↔ rank ↔ bond1 IP： .149/rank0/192.168.0.72（跑 API，不带 headless）；.134/rank1/.33；.209/rank2/.50；.163/rank3/.38（rank1-3 带 `--headless`）。

Step 0：每节点统一 HCCL TLS：`hccn_tool -i N -tls -s enable 0`（八卡全关），`-tls -g` 校验全 `switch[0]`。

Step 1：起容器（每节点）

```
docker run -d --name vllm --net=host --shm-size=1500g \
  --device /dev/davinci0 ... --device /dev/davinci7 \
  --device /dev/davinci_manager --device /dev/devmm_svm --device /dev/hisi_hdc \
  -v /usr/local/dcmi:/usr/local/dcmi -v /etc/hccn.conf:/etc/hccn.conf \
  -v /usr/local/Ascend/driver:/usr/local/Ascend/driver -v /data:/data \
  -e HCCL_OP_EXPANSION_MODE=AIV -e TASK_QUEUE_ENABLE=1 -e HCCL_BUFFSIZE=512 \
  -e VLLM_ASCEND_ENABLE_FLASHCOMM1=1 -e PYTORCH_NPU_ALLOC_CONF=expandable_segments:True \
  quay.io/ascend/vllm-ascend:v0.21.0rc1 sleep infinity
# 装 sitecustomize.py 到 site-packages(注入 DP 会剥离的 env, 见问题表)
docker exec vllm bash -c 'cp /data/sitecustomize.py $(python3 -c "import site;print(site.getsitepackages()[0])")/'
```

Step 2：起 serve（rank0 先起，sleep 4 后 rank1-3）——完整命令

```
# 容器内 env
export HCCL_IF_IP=<本机bond1> IFNAME=bond1
export GLOO_SOCKET_IFNAME=$IFNAME TP_SOCKET_IFNAME=$IFNAME HCCL_SOCKET_IFNAME=$IFNAME
export HCCL_BUFFSIZE=512 HCCL_OP_EXPANSION_MODE=AIV TASK_QUEUE_ENABLE=1
export VLLM_ASCEND_ENABLE_FLASHCOMM1=1 ACL_OP_INIT_MODE=1
export PYTORCH_NPU_ALLOC_CONF=expandable_segments:True
export HCCL_CONNECT_TIMEOUT=7200 ASCEND_CONNECT_TIMEOUT=10000 VLLM_RPC_TIMEOUT=1800000
export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libjemalloc.so.2

vllm serve /data/models/DeepSeek-V4-Pro-w4a8-mtp \
  --host 0.0.0.0 --port 10010 --api-key <key> \
  $([ $RANK != 0 ] && echo --headless) \
  --max-model-len 1048576 --max-num-batched-tokens 2048 --max-num-seqs 512 \
  --served-model-name ds-v4 --gpu-memory-utilization 0.93 --block-size 128 \
  --data-parallel-size 4 --tensor-parallel-size 8 --data-parallel-size-local 1 \
  --data-parallel-start-rank $RANK --data-parallel-address 192.168.0.72 \
  --enable-expert-parallel --quantization ascend \
  --enable-chunked-prefill --enable-prefix-caching \
  --tokenizer-mode deepseek_v4 --tool-call-parser deepseek_v4 \
  --enable-auto-tool-choice --reasoning-parser deepseek_v4 --async-scheduling \
  --safetensors-load-strategy prefetch \
  --model-loader-extra-config '{"enable_multithread_load":"true","num_threads":128}' \
  --speculative-config '{"num_speculative_tokens":1,"method":"mtp","enforce_eager":true}' \
  --additional-config '{"enable_cpu_binding":true,"enable_shared_expert_dp":true,"multistream_overlap_shared_expert":
      true,"ascend_compilation_config":{"enable_npugraph_ex":true,"enable_static_kernel":false}}' \
  --compilation-config '{"cudagraph_mode":"FULL_DECODE_ONLY","cudagraph_capture_sizes":[1,8,16,32,64,128]}'
```

Step 3：冒烟测试 `curl http://192.168.0.72:10010/v1/chat/completions`（返回 200 即成功）。

4 关键调整参数（及为什么）

参数	值	为什么
`max-model-len`	1048576	1M 上下文（真实生产场景）
`gpu-memory-utilization`	0.93	1M KV(16.6GB) 装下 + 激活余量；0.97 会并发 OOM
`max-num-batched-tokens`	2048	配 util 0.93 留激活余量
`max-num-seqs`	512	高并发上限

DP4 × TP8 + EP	—	32 卡 = 4 数据并行 × 8 张量并行，专家并行 EP32
`quantization`	ascend	W4A8 量化
`speculative-config`	mtp	MTP 投机解码（draft 头 enforce_eager）
`compilation-config`	FULL_DECODE_ONLY	**图模式（命门，见问题表）**
`model-loader-extra-config`	num_threads 128	128 线程并行加载（治加载慢，官方 A2）
`additional-config`	见命令	cpu_binding + shared_expert_dp + multistream（官方

5 遇到的问题与解决

问题	原因	解决
建 MC2 专家组报 `error 1 / EI0016 tls invalid`	HCCL TLS 跨节点不一致（某节点八卡 enable 其余 disable）	`hccn_tool -i N -tls -s enable 0` 全关统一
TPOT 646ms（慢 10×）	用了 `--enforce-eager`，61 层逐 op 同步派发	**去 enforce-eager，开 FULL_DECODE_ONLY 图模式 → TPOT 108ms**
跨节点 gloo 回环 127.0.0.1 失败	vLLM v1 DP 在 engine-core spawn 时剥离环境变量	`sitecustomize.py` 放 site-packages，python 启动自动注入 `*_SOCKET_IFNAME=bond1` 等
并发即 `NPU out of memory`	util 0.97 下 1M KV 占满只剩 136MB	util 0.93 + max-num-batched-tokens 2048
`enable_kv_nz` 启动崩	V4-Pro 的 Compress-4-Attention 非纯 MLA	不能用 enable_kv_nz
engine init 5 分钟超时崩	`enable_balance_scheduling` 等前端 5min < V4+MTP 加载 7min	不能用 enable_balance_scheduling
prefix cache 命中率恒 0%	V4 Compress-4-Attention 不支持块级前缀复用（官方 A2 教程也 `--no-enable-prefix-caching`）	已知限制，生产无前缀复用收益
加载慢 10 分钟	漏官方 model-loader 多线程	加 `num_threads:128` → safetensors 加载快约 3×
重部署崩 `Free memory 2.23GiB`	pkill 后残留 worker 卡死占住 NPU 显存（63GB 不释放）	`docker restart vllm` 清干净（保留 sitecustomize/env），显存释放到 3.4GB

6 性能

单流 TPOT 108ms（图模式，eager 时 646ms）。**最大 RPM = 1056 @ 并发 192**（input 256 / output 16，e2e 8.5s、TTFT 3.6s，本地发起压测）。并发扫描：c128 = 507、**c192 = 1056**、c256 = 538（过载，TPOT 暴涨）。解码 comm-bound ≈115 tok/s（长解码，瓶颈是跨节点 EP all-to-all，解码时 NPU 空等）；RPM 由 input/prefill 长度主导（短输入冲高 RPM）。