第一章　模型相关参数定义

第1条　Benchmark表达方式

在部署方案优化问题中，benchmark 的评估需要明确以下参数：

模型（Model）：由产品方确定，包括模型架构和参数量（Size）

输入输出长度（SeqLen）：通常由产品方确定，包括输入序列长度（Input SeqLen）和输出序列长度（Output SeqLen）

量化方式（Dtype）：通常由产品方确定，但可以协助确定最优量化方案，用 W[x]A[y] 表示（例如 W4A16 表示 weight 为 4bit，activation 为 16bit）

Batch Size（BS）：**主要评估变量**，根据前面确定的模型、参数量、输入输出长度、量化方式等参数，通过数学建模分析确定最优的 Batch Size

Benchmark 命名规则：

参考 LLM 和 CV 模型的性能指标表达方式，benchmark 命名遵循以下规则：

1) **LLM 模型 TPS：** `TPS[xxx]-[Model]-[Size]-S[SeqLen]-O[SeqLen]-W[x]A[y]-B[BS]`
   a) 示例：`TPS001-Llama2-70B-S4K-O1-W4A16-B16`
   b) 说明：Llama2-70B模型使用W4A16精度，在BS为16，输入SeqLen为4K，输出SeqLen为1时的TPS
2) **LLM模型FTL：** `FTL[xxx]-[Model]-[Size]-S[SeqLen]-W[x]A[y]-B[BS]`
   a) 示例：`FTL001-Llama2-70B-S4K-W4A16-B1`
   b) 说明：Llama2-70B模型使用W4A16精度，在BS为1，输入SeqLen为4K时的FTL
3) **CV模型FPS：** `FPS[xxx]-[Model]-[Version]-R[Resolution]-W[x]A[y]-B[BS]`
   a) 示例：`FPS001-Resnet50-V2-R224x224-W8A8-B4`
   b) 说明：Resnet50-V2模型权重和激活都使用8bit精度，在BS为4，分辨率是224x224时的FPS

评估重点：

在部署方案优化中，我们主要根据产品方确定的模型、参数量、输入输出长度、量化方式等固定参数，通过数学建模分析来评估和优化 **Batch Size**，以在满足 SLO 约束的前提下实现 TPS per Chip 的最大化。

第2条　DeepSeek-V3 配置解析

下面按功能块解释 DeepseekV3 配置，重点放在与 MoE（混合专家）路由与负载相

关的参数，同时也覆盖与注意力、尺寸、量化等有关的关键项。以下参数均来自HuggingFace DeepSeek-V3 模型卡中的`config.json`文件。可以按相同方法从其他模型的 HuggingFace 页面（如 Llama、Qwen 等）获取对应的`config.json`，为其他大语言模型制定部署方案提供参数。

1) 模型结构与尺寸
   a) hidden_size: 7168
      模型的主隐层维度（Transformer层的特征维度）。
   b) intermediate_size: 18432
      非 MoE 的标准前馈（FFN）层的中间维度（通常对应第一个线性层的扩展维度）。
   c) moe_intermediate_size: 2048
      MoE专家的FFN中间维度。说明专家的每个 FFN 比主干 FFN 要小很多，靠稀疏并行堆叠来提升总容量。
   d) num_hidden_layers: 61
      Transformer层数。
   e) num_attention_heads: 128，num_key_value_heads: 128
      注意力的总头数以及KV头数。KV与Q一致（无GQA缩减）。
   f) qk_nope_head_dim: 128，qk_rope_head_dim: 64，v_head_dim: 128
      注意力中各分量的头维度：Q/K的非ROPE部分维度、ROPE部分维度，以及V的头维度。QK被拆分为 NOPE 与 ROPE 两段。
   g) initializer_range: 0.02
      参数初始化范围（通常是正态/均匀的尺度）。
   h) rms_norm_eps: 1e-06
      RMSNorm 的数值稳定性参数。
   i) vocab_size: 129280，tie_word_embeddings: false
      词表大小；不共享输入/输出嵌入权重。
2) 注意力与位置编码
   a) attention_bias: false
      注意力线性层是否使用bias。
   b) attention_dropout: 0.0
      注意力dropout，设为0以提高吞吐（推理或大模型常见）。
   c) rope_theta: 10000
      ROPE的基础角频参数。
   d) rope_scaling: { type: "yarn", factor: 40, beta_fast: 32,

beta_slow: 1, mscale: 1.0, ... }

使用 YaRN 风格的 ROPE扩展与缩放，以支持超长上下文。factor=40 表示对原始最大长度的扩展；beta_fast/beta_slow 是 YaRN 的双速缩放参数；original_max_position_embeddings: 4096 指原始设计长度；max_position_embeddings: 163840 为最终支持长度（4096×40 ≈ 163,840）。

e) max_position_embeddings: 163840

最大位置长度，与上面一致，支持极长上下文。

3) MoE总体结构

a) n_routed_experts: 256

可被路由的稀疏专家数量（不含共享专家）。

b) n_shared_experts: 1

共享专家（dense expert），所有token都可以走它，相当于“保底”路径，常用于稳定训练和提升鲁棒性。

c) moe_layer_freq: 1

MoE层出现的频率。为1通常意味着每一层都是MoE（或每层的FFN部分是MoE版本），而不是每隔N层一个MoE。

d) routed_scaling_factor: 2.5

对经由专家路径的输出进行缩放的因子。一般用于匹配稀疏路由带来的统计幅度变化，防止MoE输出的方差与dense路径不一致。

4) MoE路由与选择策略（关键）

a) num_experts_per_tok: 8

每个token选择的专家数（Top-k）。这里是Top-8，相比常见Top-1/Top-2更大，能显著平滑负载，但计算与通信成本更高。

b) topk_group: 4

将专家分为若干组（group），先在组内或跨组做Top-k的策略。topk_group=4 表示把256专家分成4个组（每组64），路由可能先选组再选组内专家，用于降低搜索成本和改善并行通信模式。

c) scoring_func: "sigmoid"

路由打分函数。用sigmoid将路由logits压到(0,1)，与softmax不同，它不强制归一到总和为1的概率分布，常配合“norm_topk_prob”做归一化。

d) norm_topk_prob: true

对Top-k选择后的权重进行归一化（例如把选中专家的权重归一为和

为1），保证融合时数值稳定。

e) topk_method: "noaux_tc"

路由方法标识：

f) "noaux" 通常表示不使用传统的负载均衡辅助损失（auxiliary load-balancing loss）。

g) "tc" 很可能指 token capacity / topk capacity 的实现路径（具体到DeepSeek V3的源码：通常代表基于容量/阈值的截断或按token计的capacity管理策略）。整体含义是使用一种不依赖aux loss，而是依靠容量和分组/排序的路由策略来实现负载控制。

h) n_group: 8

与 topk_group 不同的分组粒度参数。DeepSeek V3中通常有两个分组相关参数：一个用于通信/并行（n_group=8，把专家或路由分在8个通信分区以优化带宽与布局），另一个用于挑选（topk_group=4）。二者共同决定路由的分区与选择路径。

i) ep_size: 1

expert parallel size。每设备上的专家并行数或专家分片数的配置。ep_size=1 表示没有跨设备的专家并行（所有专家可能在同一设备视图或采用其它并行），在实际分布式运行中这个参数通常与训练框架（如Megatron/DeepSpeed）结合使用。

5) MoE行为的其他辅助项

a) first_k_dense_replace: 3

训练或推理早期/前几层使用dense FFN替代MoE（或强制走共享专家），以稳定路由与梯度。这有助于避免训练初期的“热门专家”极化问题。值为3表示前3层用dense。

b) topk_method 与 capacity 的关系

虽未直接给出 capacity_factor，但“noaux_tc”暗示采用基于容量的截断/重排策略，常见做法是每个专家有每步的token容量上限，超出则丢弃或转移到共享专家。这能在没有aux loss的情况下维持负载可控。

6) 并行与缓存

a) use_cache: true

推理时缓存KV以加速自回归生成。

b) attention_bias: false, attention_dropout: 0.0

这两项在推理友好配置中常见，减少额外开销。

7) LoRA相关（注意力权重低秩分解，特定于DeepSeek V3）

a) q_lora_rank: 1536

对Q投影采用LoRA的秩。

b) kv_lora_rank: 512

对K/V投影采用LoRA的秩。

这些用于降低参数/计算或更易微调，同时保留较强表达能力。

DeepSeek V3中Q/K/V有分头维度与ROPE/NOPE拆分，配合LoRA可以更灵活。

8) 量化与数值精度

a) torch_dtype: "bfloat16"

主计算精度为BF16。

b) quant_method: "fp8", fmt: "e4m3"

使用FP8（E4M3格式）进行权重或激活的量化以提升吞吐与减小显存。

c) activation_scheme: "dynamic"

动态激活量化策略（按批或按层动态调整尺度）。

d) weight_block_size: [128, 128]

权重量化的块大小（影响scale的粒度与吞吐）。

e) scale_fmt: "ue8m0"

量化scale的数据格式。

这套配置表明模型走混合精度：主算BF16，但某些权重/激活用FP8量化以进一步优化性能。