第三章 算力的基石：超越规模的极限

论证AI发展正从“规模驱动”转向“效率驱动”，并阐明自主全栈AI生态的战略必然性。

密集Transformer的三大计算瓶颈

以“越大越好”为核心的规模法则正因其不可持续的成本而走向尽头，其根源在于传统密集Transformer架构固有的三大计算瓶颈。

🤯

计算复杂度瓶颈

自注意力机制的计算量随序列长度二次方增长（$O(N^2)$），处理长文本成本极高。

🏋️

密集激活瓶颈

每个输入都需激活模型全部参数，导致推理成本与参数量直接挂钩，模型越大越慢。

📚

知识存储瓶颈

所有知识都存储于模型参数中，导致知识静态、可能过时，且容易产生“幻觉”。

AI的新轨迹：效率型架构的崛起

面对上述瓶颈，业界正转向以Mamba、MoE、RAG为代表的效率型架构，它们分别从不同维度对Transformer进行了战略性的“解绑”。

Mamba (SSM)

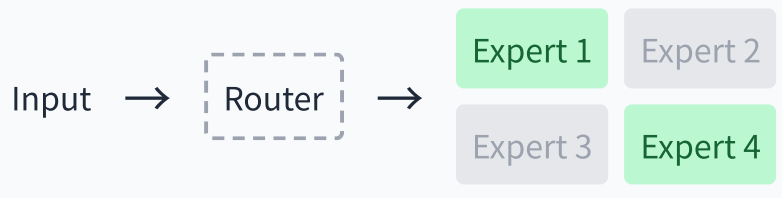
通过引入“选择机制”的状态空间模型，将计算复杂度从 $O(N^2)$ 降至 $O(N)$，在高效处理超长序列的同时，保持了强大的性能，解决了“计算复杂度”瓶颈。

Mamba 线性处理流程

Input (t) → **选择性更新** → State (t) → Output (t)

将历史信息压缩到一个循环更新的“状态”中，实现线性计算。

MoE 稀疏激活示意图



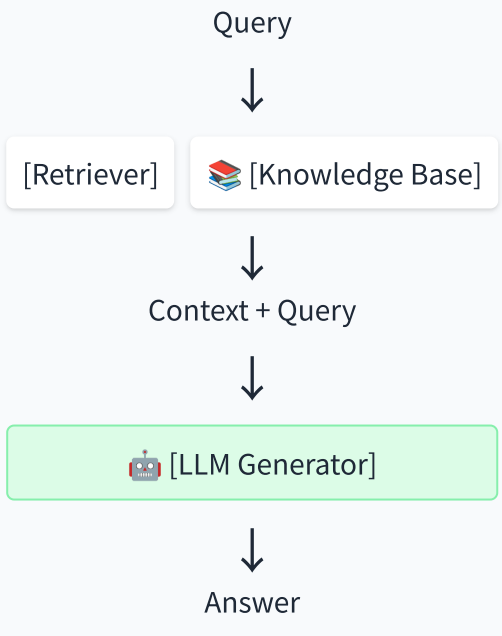
专家混合 (MoE)

通过“稀疏激活”策略，每个输入仅由一小部分“专家”网络处理，成功解耦了模型的总参数量与单次推理的计算量，解决了“密集激活”瓶颈。

检索增强生成 (RAG)

将知识外置于可动态更新的数据库中，模型在生成时先“检索”相关信息再作答。这让模型更轻量、内容更可信且可实时更新，解决了“知识存储”瓶颈。

RAG 工作流程



战略的必然：自主全栈生态对比

要将效率型架构的理论优势转化为实际性能，必须拥有软硬件协同设计的全栈技术生态。这已成为地缘政治背景下的关键博弈点。

NVIDIA/CUDA 生态系统

应用框架: PyTorch, TensorFlow...

使能库: TensorRT, cuDNN, cuBLAS...

核心API: CUDA

硬件: GPU (内置 Tensor Core)

华为昇腾 (Ascend) 生态系统

应用使能: MindX SDK, ModelArts

AI框架: MindSpore

计算架构: CANN (类似CUDA+cuDNN)

硬件: NPU (达芬奇架构, 内置 AI Core)

对比维度	NVIDIA 生态	华为昇腾 生态
核心战略	横向开放，构建行业事实标准。	垂直整合，自主可控。
核心优势	硬件性能领先，拥有全球最庞大的开发者生态。	软硬件深度协同优化潜力巨大，符合国家战略安全。