LLM MoE Multicast use case

https://datatracker.ietf.org/doc/draft-zhang-rtgwg-llmmoe-multicast/

IETF 123 Madrid

Sandy Zhang
(ZTE)

LLM MoE Introduction

- MoE (Mixture of Experts) is a popular architecture in LLM (Large Language Model), where multiple experts are activated during the computation and only part of the parameters are used for each expert.
- MoE brings a more cost-effective training and inference method. It enables the deployment of larger models at almost the same cost. More and more LLMs adopt MoE, such as DeepSeekV3, Llama 4, Mixtral, Qwen3, etc.
- Using the gating function, a token is sent to multiple selected experts. This is a typical multicast use case.

The MoE deployments

- Mixtral-8x7B has 8 experts, two experts are activated at a time.
- In Llama 4, Scout has 16 experts, Maverick has 128 experts, two experts are activated at a time.
- DeepSeekV3 has 256 experts, 9 of which are activated at a time: one shared expert and eight routing experts.

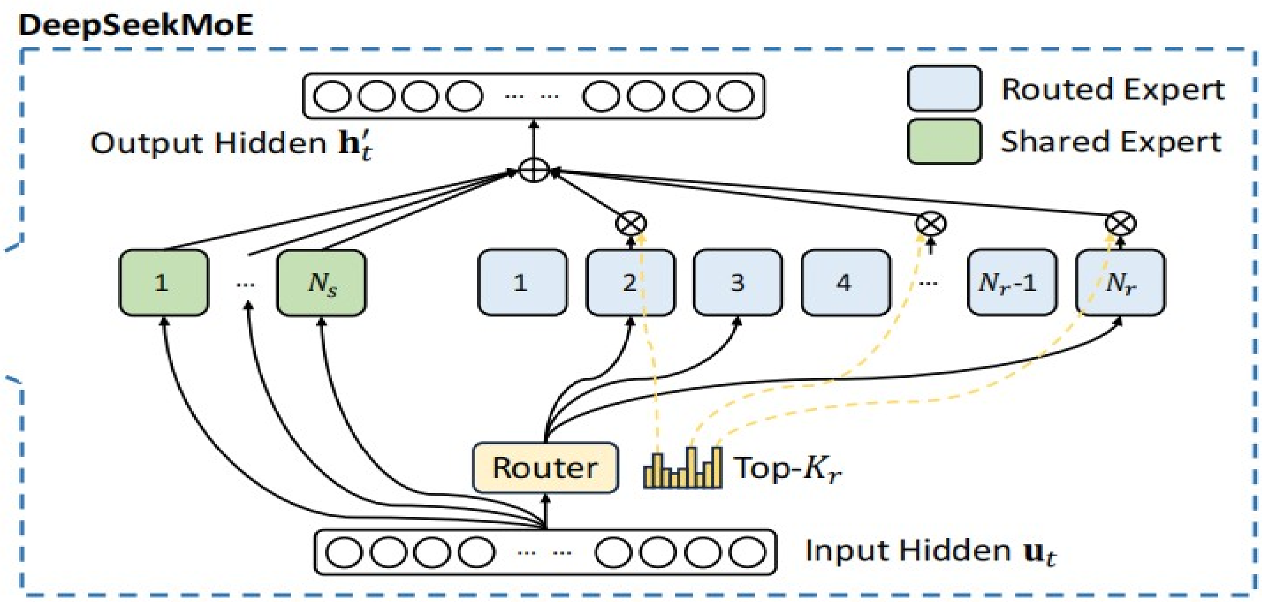
MARKET TRENDS – GENERATIVE AI COMPUTING

Mixture of expert (MoE) vs. dense models

	Google Gemini 1.5 Pro	xAI Grok-1	Mistral 8x22B	DeepSeek-v2	OpenAI GPT-4o	META Llama-3.1 405B	Claude-3.5 Sonnet	DeepSeek-v3	Qwen2.5 72B
Launch date	Feb. 2024	Mar. 2024	Apr. 2024	May. 2024	May. 2024	Jul. 2024	Oct. 2024	Dec. 2024	Jan. 2025
Country									
Architecture	MoE	MoE	MoE	MoE	Likely MoE	Dense	-	MoE	Dense
# of activated parameters	-	78B	39B	21B	-	405B	-	37B	72B
# of total parameters	-	314B	141B	236B	~1.8T	405B	-	671B	72B
Ratio param:activated	-	4x	3.6x	11.2x	-	1x	-	18x	1x
# of layers	-	64	56	27	-	126	-	61	80

Mixture of expert (MoE) architectures are now widely adopted in many recent LLM AI models.
During inference, MoEs activate only a fraction of their parameters (activated parameters ~ experts) to determine each output token, significantly reducing the number of computations required on the processor and thus lowering the overall computing demand.
Consequently, the fewer parameters activated, the fewer computations the processor needs to perform. However, a high number of layers tends to reduce the model's efficiency, as inevitable networking slowdowns occur between the different processors at large scale -> it's more difficult to parallelize calculations.

LLM MoE (DeepSeekV3)



- Using the gating function, tokens are sent to multiple selected experts.
- Since experts need to be balanced, it is difficult to always use the same set of experts.
- Experts are distributed across different GPUs and nodes, so it is difficult to restrict the selected experts to one node.
- The communication of EP (Expert parallelism) may span leaf switches or even spine switches.

- A token is sent to multiple selected experts.
- This is a typical multicast use case, and just one of many.

The motivation

- Use network multicast technology to assist LLM multicast implementation
  - Reduce the pressure on the source GPU
  - Reduce the possibility of congestion caused by bandwidth between nodes

- Finding the possibility of using multicast method for use cases inside and outside nodes

GAP

- Currently, the sending methods within the card (memory reading), between cards (Nvlink), and between nodes (RDMA) are inconsistent.
- Coordination between large model coding, communication library, network card, and network layer
- Location and encapsulation of multicast
- Management by experts (MoE scenario)

Open Questions

- Do we need one multicast technology to solve all multicast use cases in LLM?

- Do we need some specific technologies for some specific use cases (e.g. MoE)?

LLM Multicast side meeting

- Tuesday 9:45-11:30 (Tomorrow morning)
- Meeting room: El Escorial, Ground Floor

- Thanks