FlowMoE: A Scalable Pipeline Scheduling Framework for Distributed Mixture-of-Experts Training

Yunqi Gao[1], Bing Hu[1,*], Mahdi Boloursaz Mashhadi[2], A-Long Jin[3], Yanfeng Zhang[4], Pei Xiao[2], Rahim Tafazolli[2], Mérouane Debbah[5]
1-Zhejiang University · 2-University of Surrey · 3-Xi'an Jiaotong-Liverpool University · 4-Northeastern University · 5-Khalifa University

Motivation

- MoE scales LLM without increasing computational costs.
- Existing works pipeline only expert + all-to-all, ignoring MHA, gating, and all-reduce.
- Ignored tasks = 30–40% of iteration time → huge inefficiency.

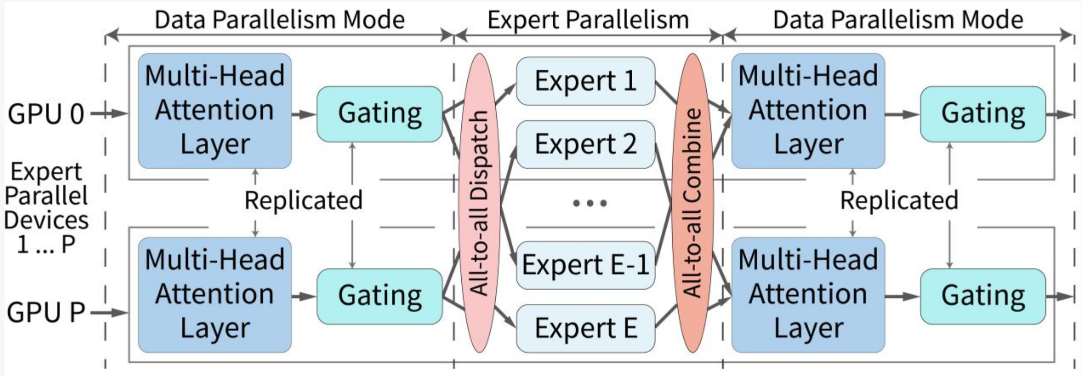


Fig. 1: Training MoE model with expert parallelism

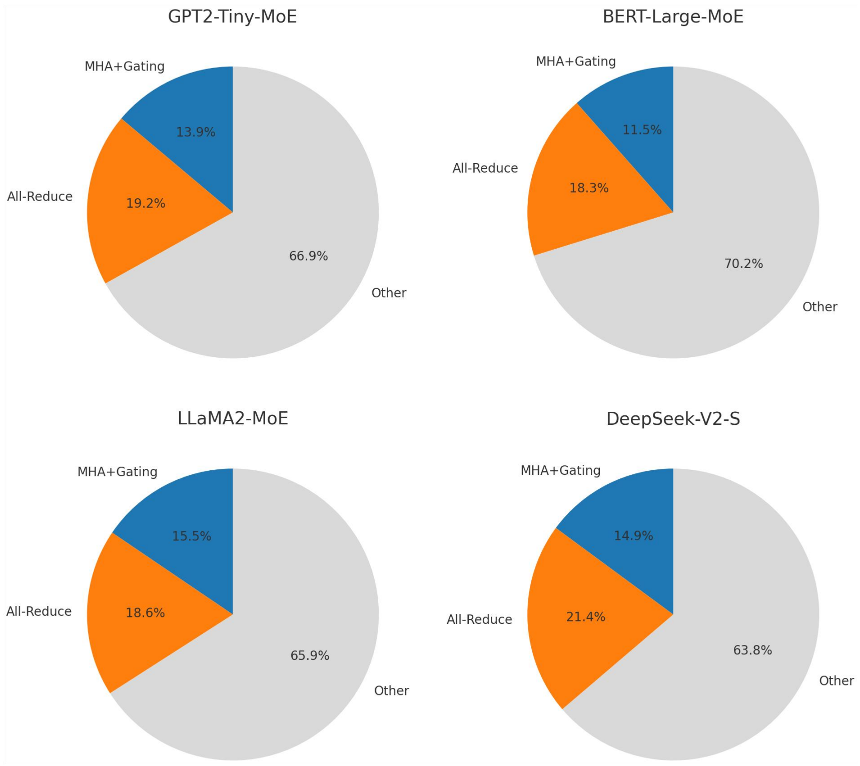


Fig. 2: One iteration time breakdown per model

NEURAL INFORMATION PROCESSING SYSTEMS

FlowMoE — Key Ideas

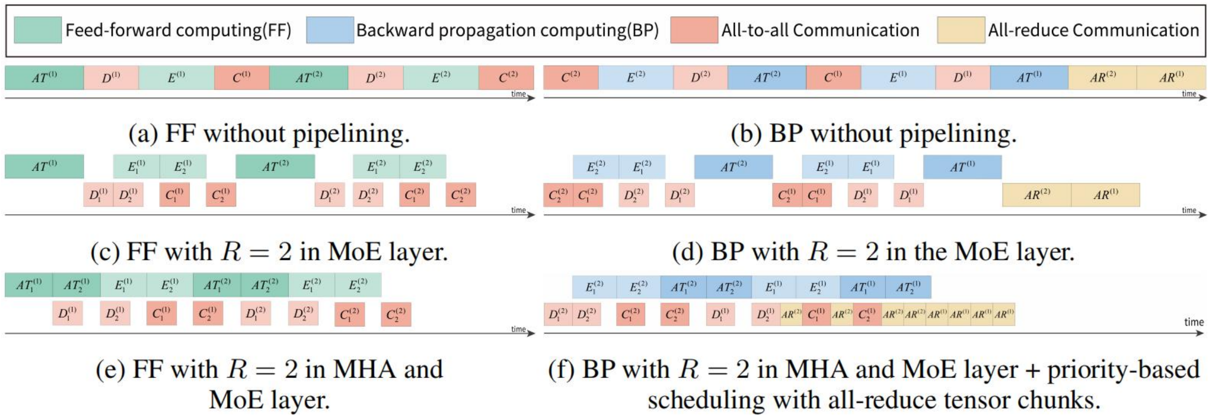


Fig. 3: Scheduling timeline of computation and communication tasks in one iteration.

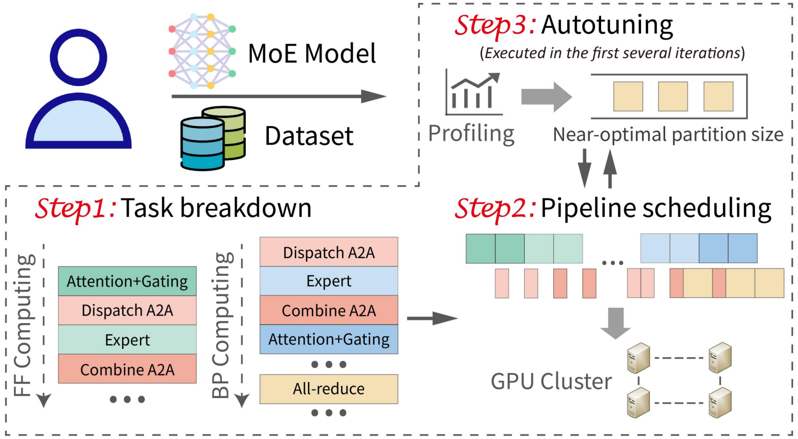


Fig. 4: FlowMoE workflow

- **Unified pipeline scheduling strategy:** Schedules MHA, gating, expert, and A2A together.
- **Priority scheduling mechanism for heterogeneous communication tasks:** Cut all-reduce chunks and execute them in all-to-all task gaps.
- **Lightweight adaptive optimizer and system integration:** Tiny Bayesian optimizer for automatic tuning. Deploying FlowMoE to the PyTorch engine.

Experimental Results

Experimental Settings

- 675 customized MoE layers, 4 real-world MoE models
- Clusters: RTX3090 (16 GPUs), RTX2080Ti (8 GPUs).

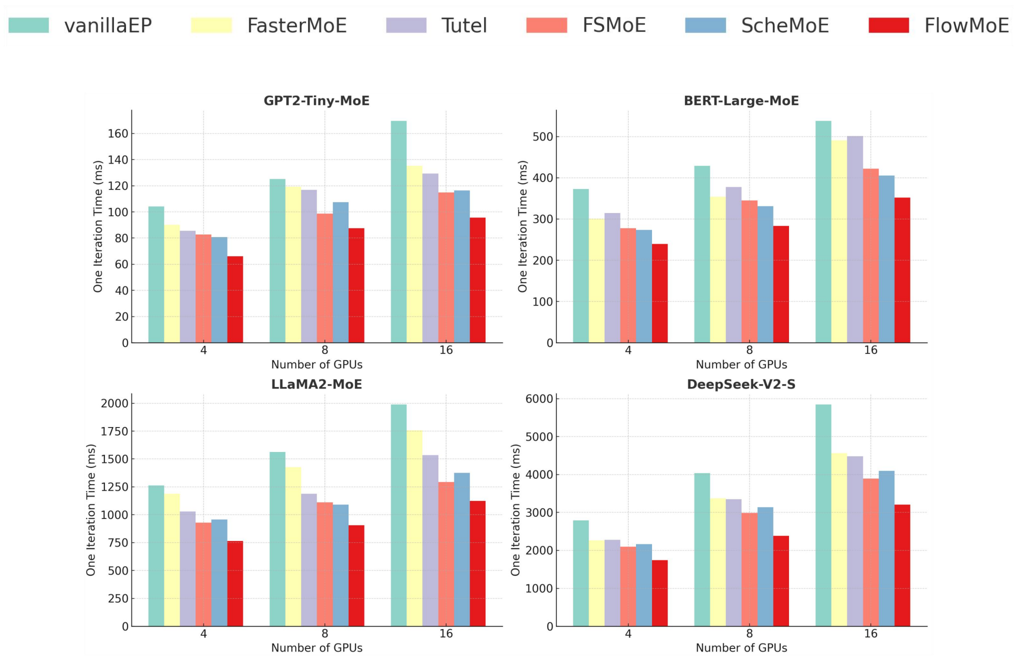


Fig. 5: Comparison of average per-iteration time in milliseconds.

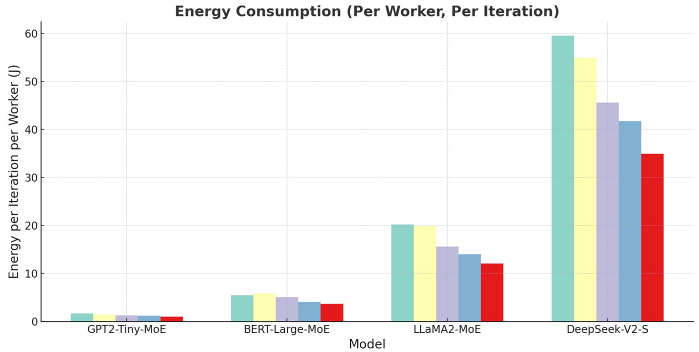


Fig. 6: Averaged per-worker energy consumption in one iteration.

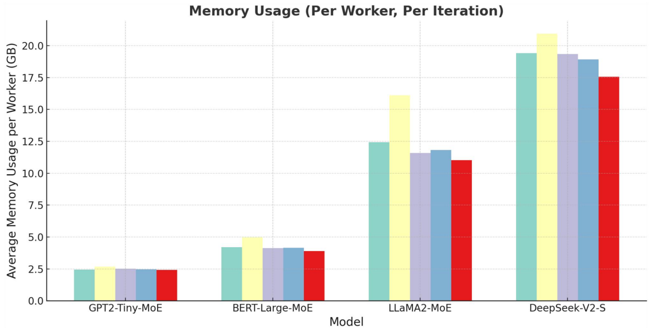
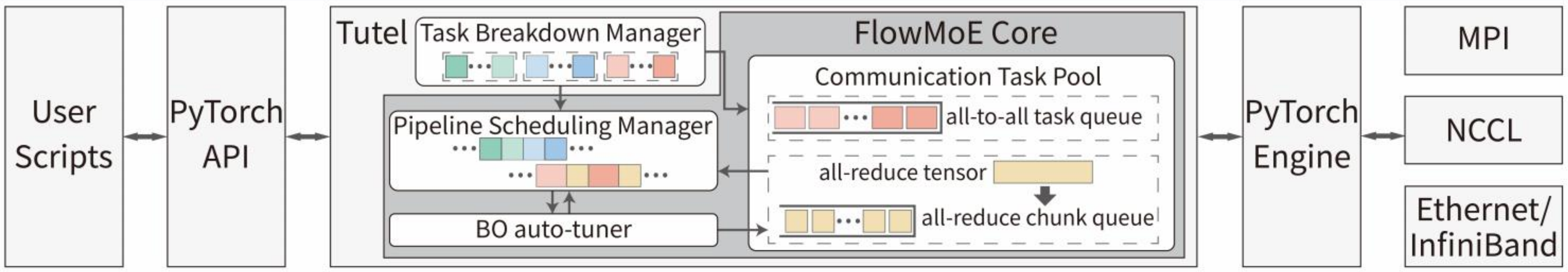


Fig. 6: Averaged per-worker memory usage in one iteration.

System Implementation & Conclusion

System Implementation



- **Framework:** Implemented on PyTorch , leveraging Tutel for optimized communication.
- **Compatibility:** Supports multiple optimization frameworks and communication stacks.

Conclusions

- The unified scheduling across all major MoE-related tasks.
- Enabling the optimal coexistence of heterogeneous communication tasks.
- Substantially advancing distributed pipeline training.
- Open-source: github.com/ZJU-CNLAB/FlowMoE