SGLang Roadmap Update

Yineng Zhang

SGLang Lead @ LMSYS Org

Outline

Review 2025 H1 roadmap

Review 2025 H2 roadmap

Feedback (Q & A)

Links:

https://github.com/sgl-project/sglang/issues/4042

https://github.com/sgl-project/sglang/issues/7736

Review 2025 H1 Highlights

Review of 2025 H1 roadmap

Focus:

Throughput-oriented large-scale deployment

Reinforcement learning training framework integration

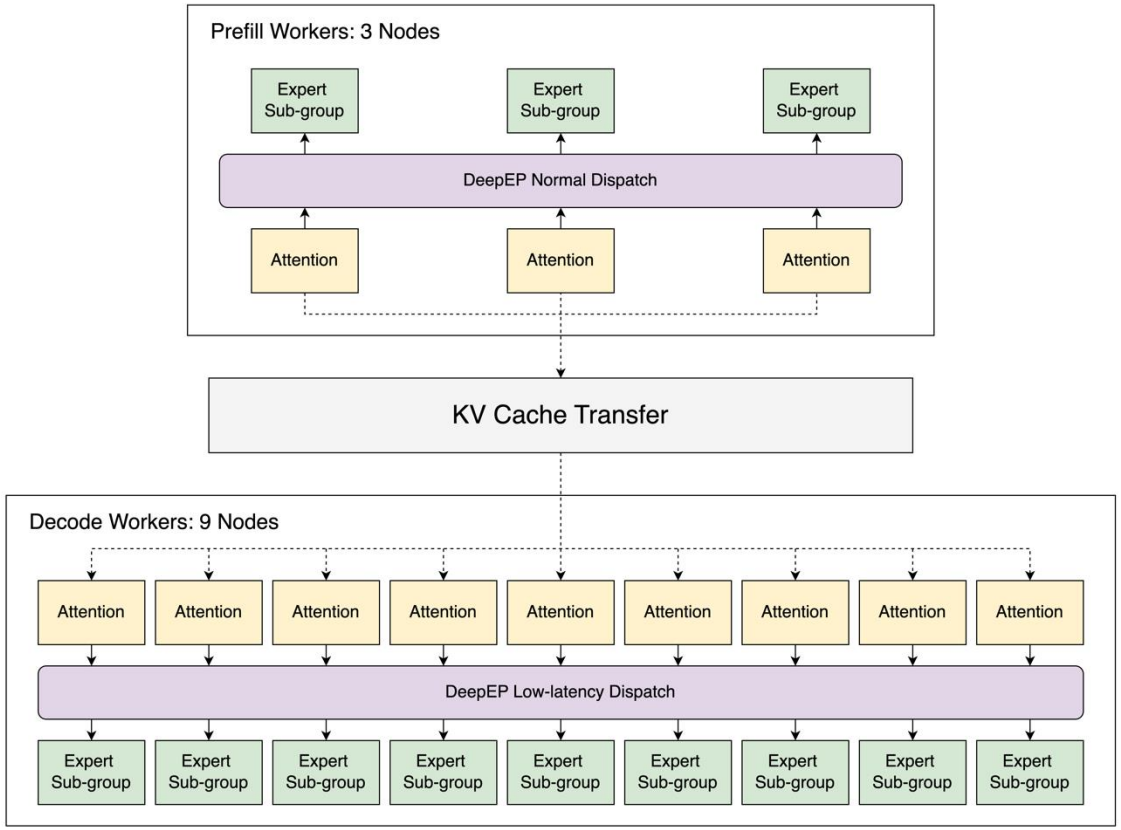
Long context optimizations

Low latency speculative decoding

Kernel optimizations

2025 H1 highlight: large-scale deployment

SGLang is **the first open-source system** that nearly match the performance of DeepSeek official blog with PD disaggregation and EP



Performance at (May. 2025)

- 52.3k input token/s/node
- 22.3k output token/s/node
- **5x cheaper than DeepSeek API price**

Reproduced by 10+ other teams

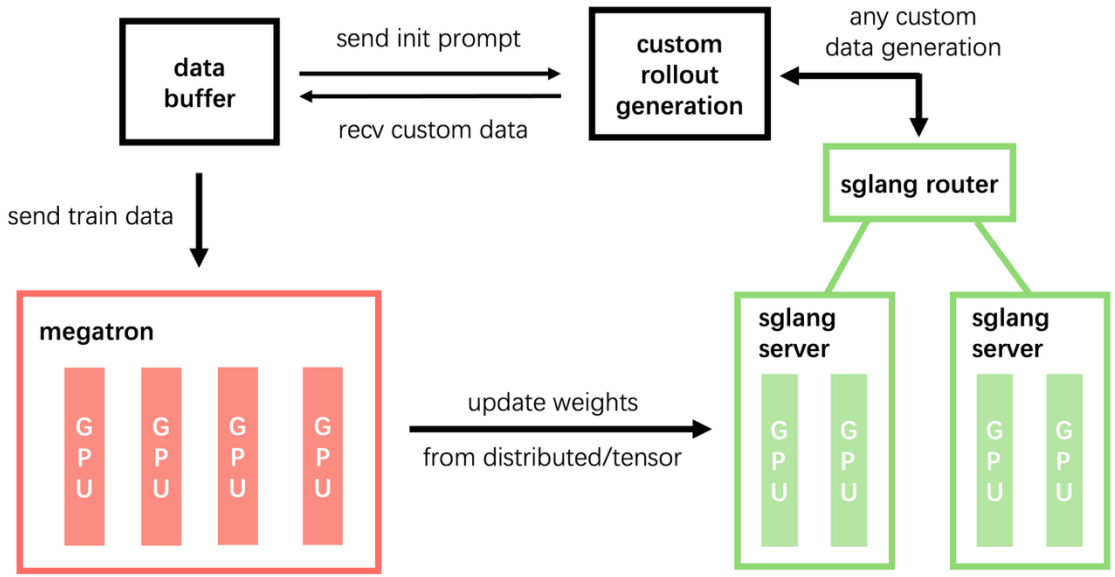
2025 H1 highlight: RL integration

SGLang-native frameworks: slime and AReaL

General integration: veRL

Case Study:

slime architecture: Megatron-LM + SGLang for extreme scalability. Used to train GLM-4.5



2025 H1 highlight: adoptions

Grow the dev community to 600+ contributors and 50+ institutions.
The default DeepSeek inference engine for 10+ companies in the first month of release.

Review 2025 H2 Roadmap

2025 H2 Roadmap

Focus

Feature compatibility and production-level reliability

Usability: simple launch scripts for large-scale deployments

Kernel optimizations for new generations of hardware

Reinforcement learning training framework integration

Distributed hierarchical KV cache system

Feature compatibility and reliability

Support performant combination of all major features

PD Disaggregation x Speculative Decoding x All kinds of parallelism x All kinds of memory pool x Overlap scheduler

Production-level reliability

- Better CI coverage
- Crash dump, replay and report

Usability

- Simple launch script for large scale deployment
- Integration with OME, sgl-router, and more

OME is a Kubernetes operator for enterprise-grade management and serving of Large Language Models (LLMs).

Use case: launch a 128-GPU cluster for deploying kimi-k2 with PD and EP with one click

Kernel optimizations

- **aiter kernels**
  - Integrate aiter attention and moe kernels
  - Close collaboration with AMD aiter

- **Communication kernels**
  - Faster allreduce on multi node
  - Faster all-to-all kernels

RL framework integration

Continue our collaboration with slime, AReal, and veRL

Goals:

- Simple recipe to finetune the full size DeepSeek/Kimi
- Faster weight sync
- Techniques to handle long tail and asynchrony
- Tools for matching numerics and logprobs with trainers

KV Cache system

Hierarchical cache

- Offloading KV cache to CPU
- Blog post coming soon

Distributed cache

- Two backends: Mooncake store, DeepSeek 3FS
- Blog post coming soon

Community building

Collaboration with AMD

- Grow 10+ code owner/committer/reviewers from AMD
- Build robust CI and merge pipeline

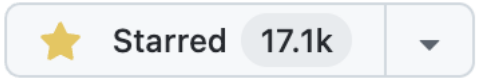
Collaboration with inference services providers

- Dedicated support and weekly sync meetings with major inference providers
- Provide Day-0 support for new models

Collaboration with broader dev communities

- Slack workspace with 3.9k developers
- Bi-weekly open dev meetings

Question & Answer



GitHub: https://github.com/sgl-project/sglang

Welcome to join our slack and bi-weekly dev meeting!