SGLang Router

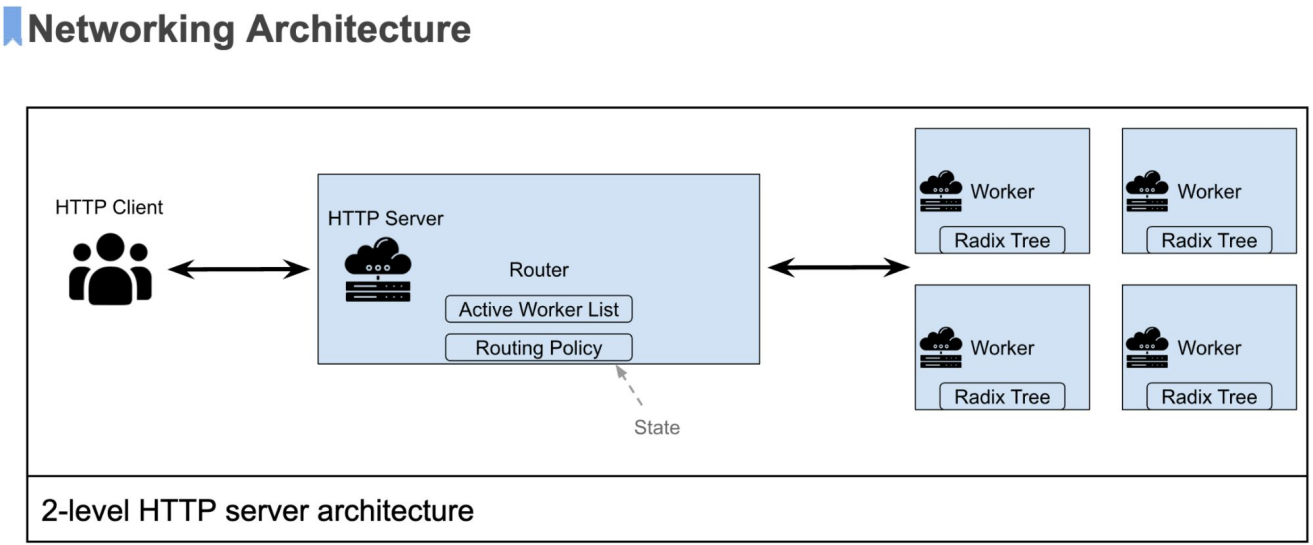
2024/11/16

Recap

- Cache-aware DP for SGLang
- **Prefix cache aware**: the requests are sent to the worker with higher cache hit rate
- **Load balancing**: no workers are overloaded or underloaded
- **Fault tolerance**: a worker can be removed and recovered without affecting the availability
- **Weight sync**: weights can be transmitted by network because the network bandwidth is higher than disk bandwidth in high-end cluster

Recap

- **Prefix cache aware**: the requests are sent to the worker with higher cache hit rate
- **Load balancing**: no workers are overloaded or underloaded



What we have done

✅ #1790:

- Decided to use rust for router
- Benchmark Python Router v.s. Rust Based Router (2x+ faster!!)
- Rust http server nearly introduces no overhead

✅ #1934:

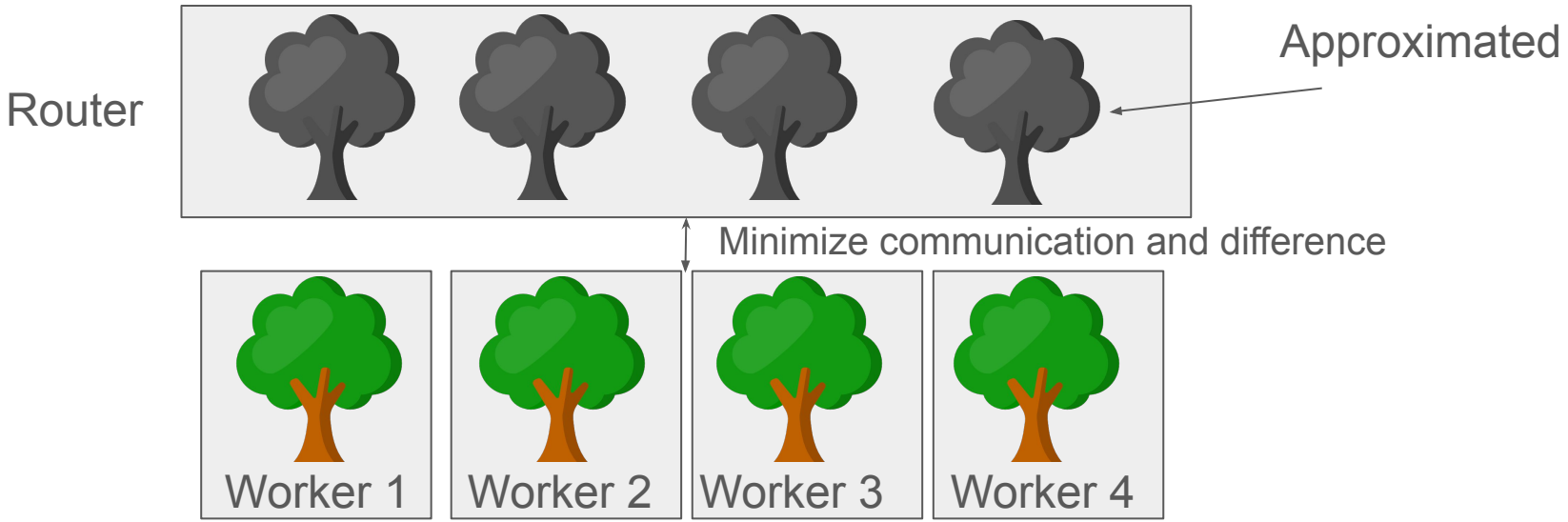
- Designed approximate tree algorithm to simulate local trees
- Implemented approx tree and achieved 4x cache hit rate and -30% latency

✅ #2002, #1999:

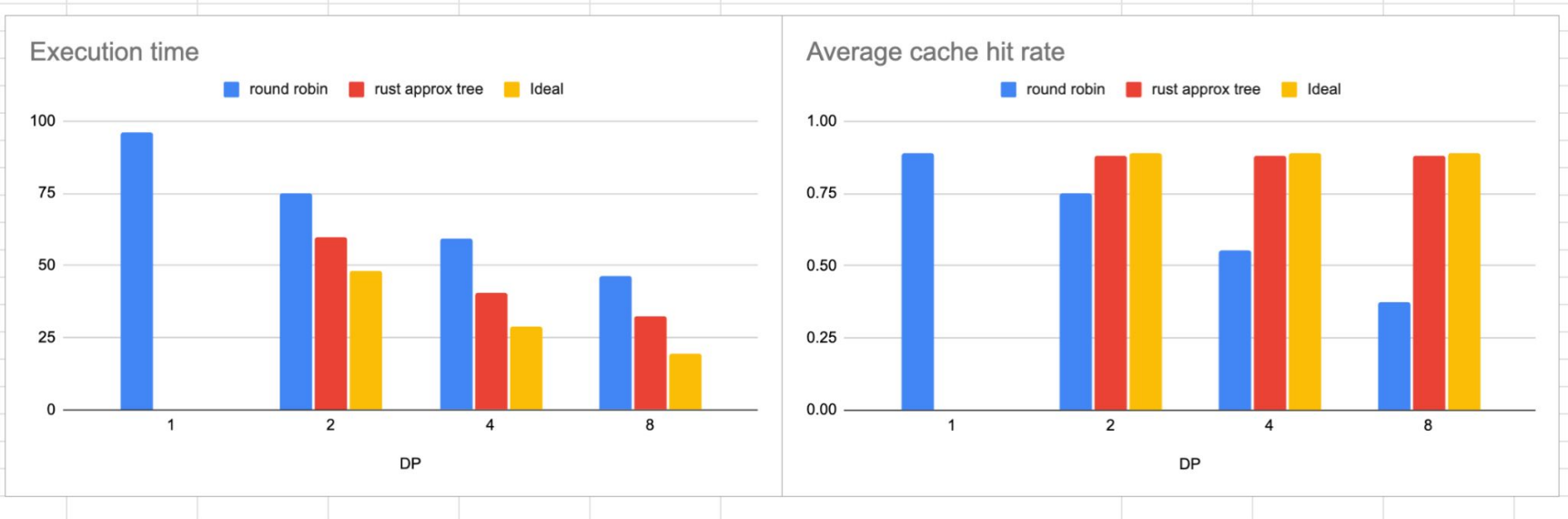
- SGLang rust infra setup (Rust testing, Python binding publishing, etc)

Technical Challenges

- Python HTTP server is very very slow
- How to minimize communication between router and worker?
- How to minimize the difference of approximate tree and local tree?
- How to design a meaningful long-prefix benchmark?



Benchmark



The difference from ideal case may be optimized by reducing concurrency overhead

Ongoing work

- Overlapped approximated tree on the router: https://github.com/sgl-project/sglang/pull/2019
- Periodic LRU leaf eviction
- Optimize performance of the concurrent overlapped tree