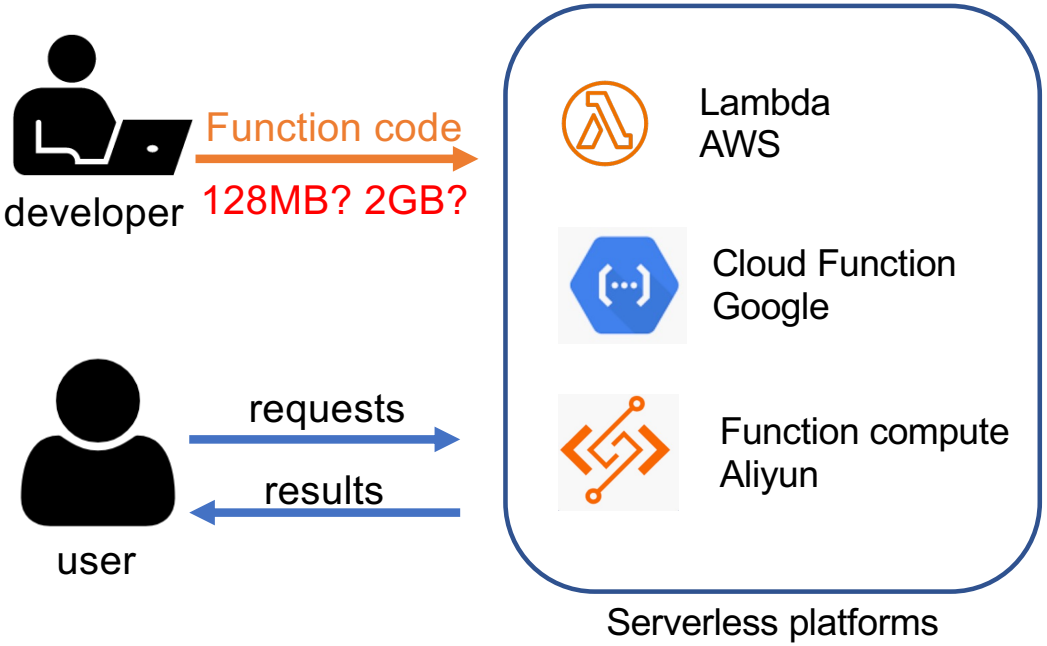
Fuyun: Bridging the Semantic Gap in Serverless Resource Provisioning via LLM Agents

Qingwen Li, Kai Lv, Mingxuan Yang, Cunchi Lv,
Zhengyu Lei, Xiao Shi, Xiaofang Zhao

Presenter: Qingwen Li

AgenticOS Workshop at ASPLOS 2026
March 23, 2026

What is Serverless Computing?



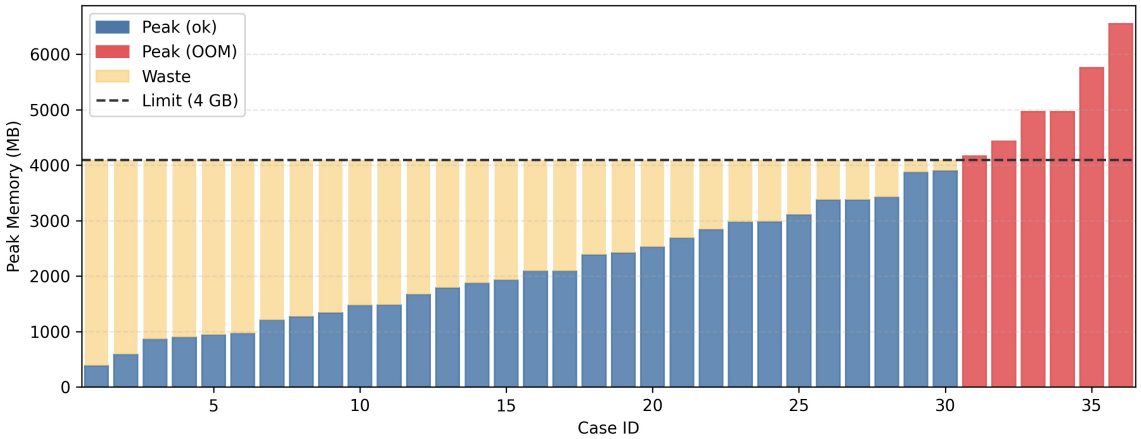
Advantage

- ✓ No server management
- ✓ Elastic scaling with demand
- ✓ Charges only for actual usage

Problem

Developers still need to
choose function-level resources.

Solution 1: Static Provisioning



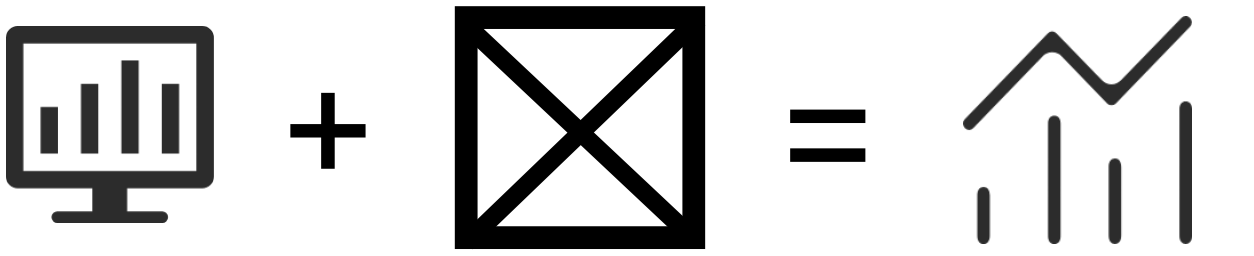
Example: applying one 4 GB memory limit to 36 different inputs

Fixed resource limit Fails

- ✓ One fixed limit cannot fit all inputs
- ✓ Small inputs waste resources
- ✓ Large inputs may fail with OOM

Resource provisioning must be input-aware, not fixed.

Solution 2: Automatic Provisioning



Monitoring data

BO / RL methods

Source code semantics

Provisioning policy

Consequence

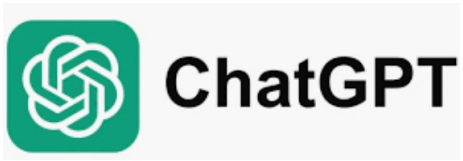
- High sample complexity
- expensive tuning overhead

Example: FaaSConf

- 31K offline samples
- 43 days to collect

Can source-code semantics help guide resource provisioning?

New opportunity with LLM



LLMs have shown strong abilities in code understanding.

- ✓ Code generation
- ✓ Debugging and fixing
- ✓ Code Explanation and Documentation
- ✓ Language Translation

LLMs make it possible to bridge source-code semantics and resource provisioning.

Challenges in LLM-Driven Resource Provisioning

Reliability
LLM outputs are not always trustworthy

Overhead
Online LLM inference would introduce latency and cost

Adaptation
Provisioning policies must evolve with runtime feedback

Key Idea: Separate Semantic Structure from Runtime Parameters

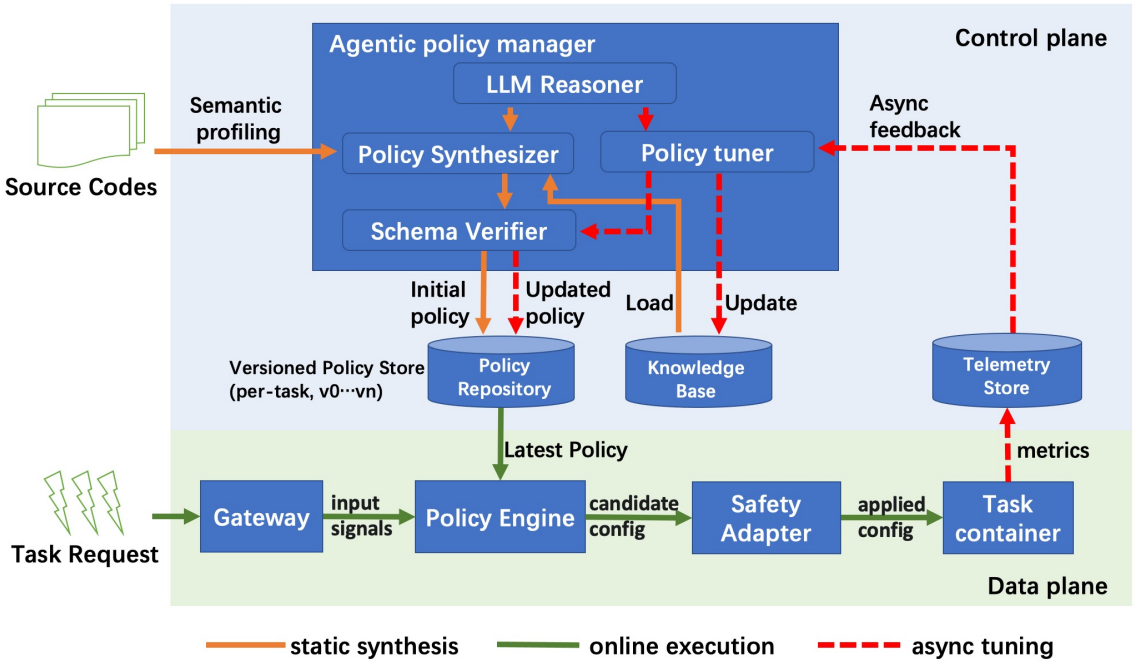
Stage 1
Build the semantic policy skeleton

Source code + LLM agents = semantic policy skeleton

Stage 2
Tune the parameters with runtime feedback

Semantic policy skeleton + Monitoring data = Deployable provisioning policy

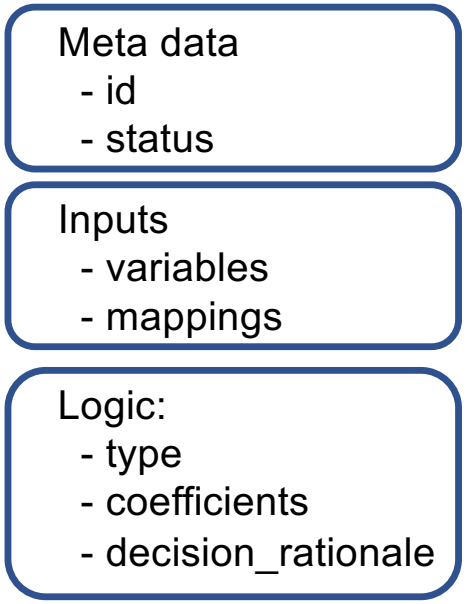
Overview of Fuyun



Three-path decomposition

- Initial policy generation offline
- Lightweight policy instance creation
- **Async tuning from metrics**

Mechanism 1: Schema-constrained Policy Representation



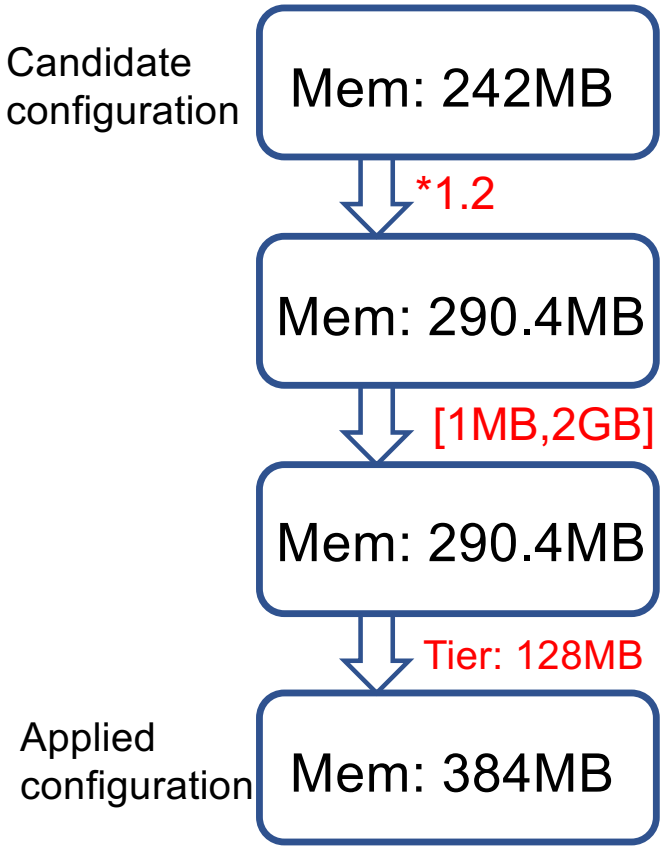
Policy template

Fuyun does not accept free-form LLM outputs; it compiles them into a strict policy schema.

Benefits

- ✓ Structured policy representation
- ✓ Verify before deployment
- ✓ Reduce hallucination risk

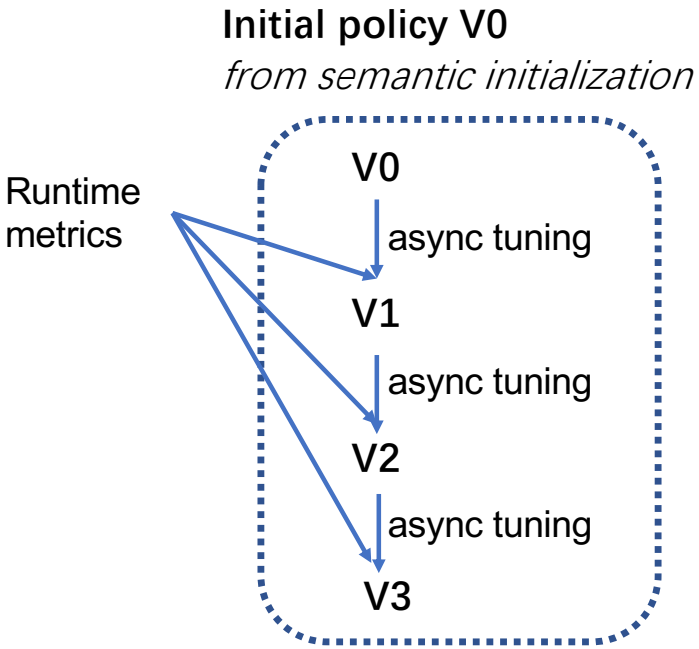
Mechanism 2: Safe, Provider-Compatible Adaptation



Fuyun does not apply candidate configuration directly.
Convert it into a safe, deployable configuration.

Add safety headroom for transient spikes
Clip to valid bounds for platform and user constraints
Snap to provider tiers for discrete resource options

Mechanism 3: Asynchronous policy parameter tuning



How tuning works

- ✓ Semantic initialization
- ✓ Metrics-driven fitting
- ✓ Asynchronous publishing

When to update policy

- ✓ On execution failure
- ✓ On large prediction error
- ✓ At periodic intervals

Evaluation: Questions and Setup

Questions：

- **Q1:** Can Fuyun produce a strong initial policy and converge quickly?
- **Q2:** Can Fuyun achieve both high reliability and low resource waste?

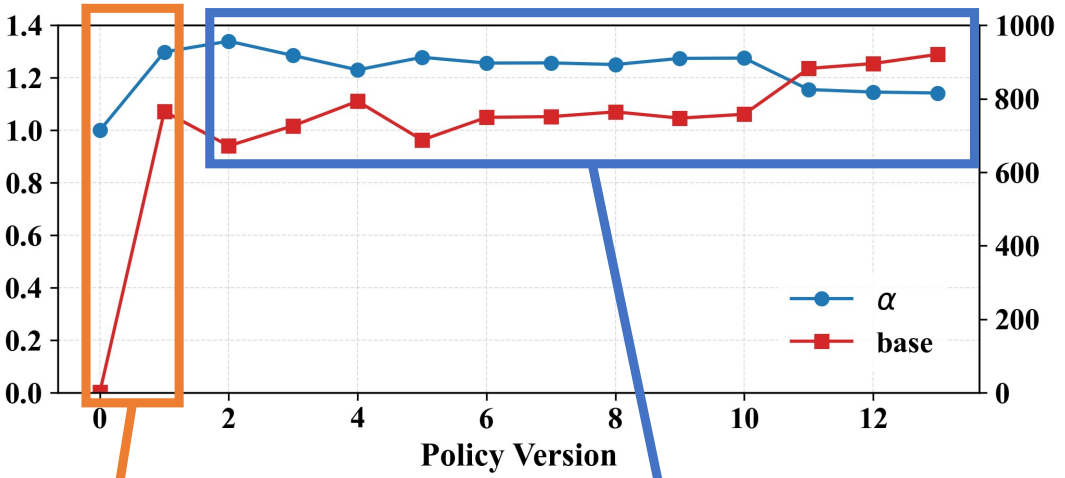
Setup

- **Benchmark:** video processing from SEBS benchmark
- **Metrics:** success rate and waste ratio
- **LLM model**: GPT-4o.

Exp1: Evolutionary Convergence

Initial policy sketch from semantics

- Model: Linear model
- Form: $y = base + \alpha \cdot$(duration·fps·width·height·3)
- Why this sketch: memory usage is highly correlated with video dimensions and frame count.



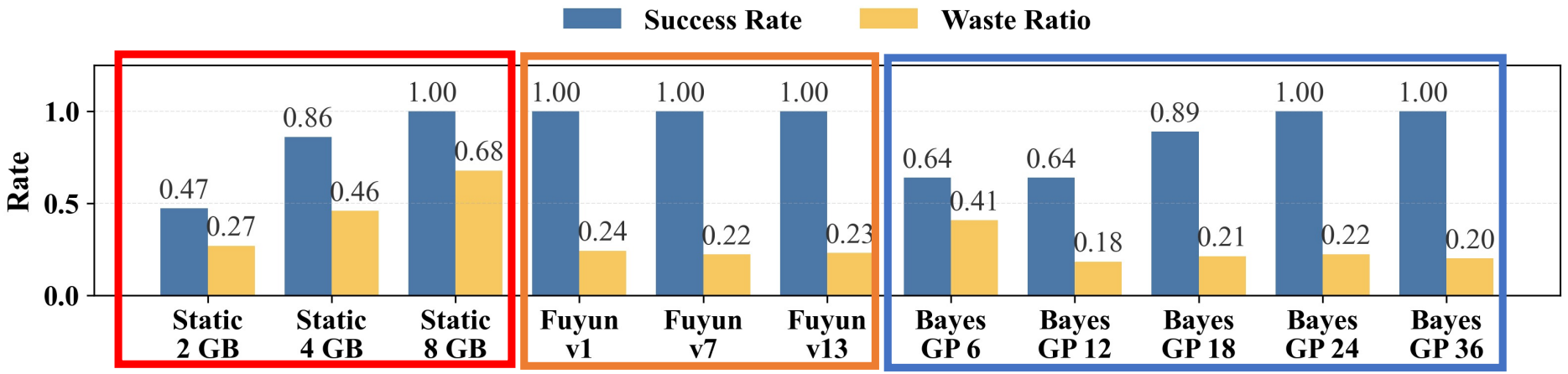
Good initial policy sketch

Fast early convergence

Stable later parameters

Good Initialization Leads to Fast and Stable Convergence

Exp2: Performance Overview



Static
Clear trade-off
High waste, more reliable

Fuyun
Full reliability
Lower waste

Bayesian Gaussian Processing
More data more reliable
Need 4× more data than Fuyun

Fuyun Achieves Full Reliability with Lower Waste

Future Work

Policy Space

- More provisioning logic choices
- Richer policy templates

Evaluation

- More baselines
- More benchmarks
- Diverse execution environments
- Ablation studies

Conclusion

- ✓ **Bring source-code semantics into resource provisioning**
- ✓ **Decouple LLM reasoning from the online execution path**
- ✓ **Achieve full reliability with lower waste and fewer samples**

Thanks !

liqingwen21b@ict.ac.cn