AI Driven Observability - Massive AWS Serverless Workflows

Adaptive Parallelism Insights - Conf 42 Observability 2025

Vamsi Praveen Karanam,
Senior Software Engineer, Amazon

The Challenge of Serverless Orchestration

Scalability Challenges

- Hidden latency spikes
- Tracing gaps
- Raising costs

Dashboard Limitations

- Aggregate metrics
- Masking branch-specific anomalies

Key Questions

- Fine-grained telemetry
- Real-time insight
- Safe Automation

Background and Related Work

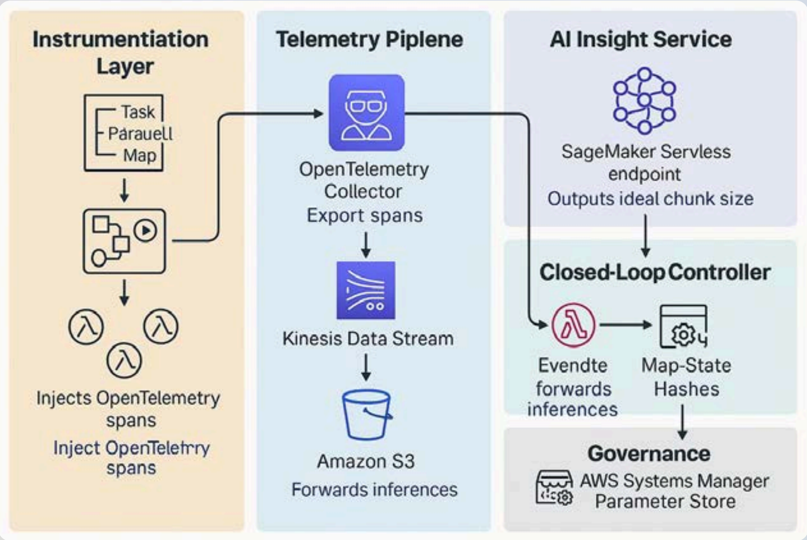
Serverless Bulk-Update Patterns

Observability Standards

AI in Cloud Optimisation

Adaptive Parallelism Insights System Architecture

- Instrumentation Layer
- Telemetry Pipeline
- AI Insight Service
- Closed-Loop Controller
- Governance



Guard-Rails in the Closed-Loop Controller

- Chunk-Change Window
- Concurrency Cap
- Rollback Trigger

These measures ensure that AI suggestions cannot destabilize production systems while still allowing for meaningful optimization. Every inference JSON is hashed and versioned in AWS Systems Manager Parameter Store, with span context IDs embedding the hash to maintain lineage between decision and outcome.

Evaluation Methodology

Workloads

- **Synthetic-Uniform (SU):** 10 TB of 1 MB records with uniform processing time.
- **Synthetic-Skewed (SS)**: 10 TB where 5% of records vary in size.
- **Live Enterprise Entity Data (LE)**: 3 TB of anonymized records.

Baselines

- Static batching (1,000 items)
- Dynamic subtree partitioning
- APIx adaptive chunking.

Metrics

- Latency: p50, p90, p95, and maximum per execution.
- Cost: absolute USD billed (Lambda plus Step Functions).
- Reliability: number of throttle errors and automatic retries.

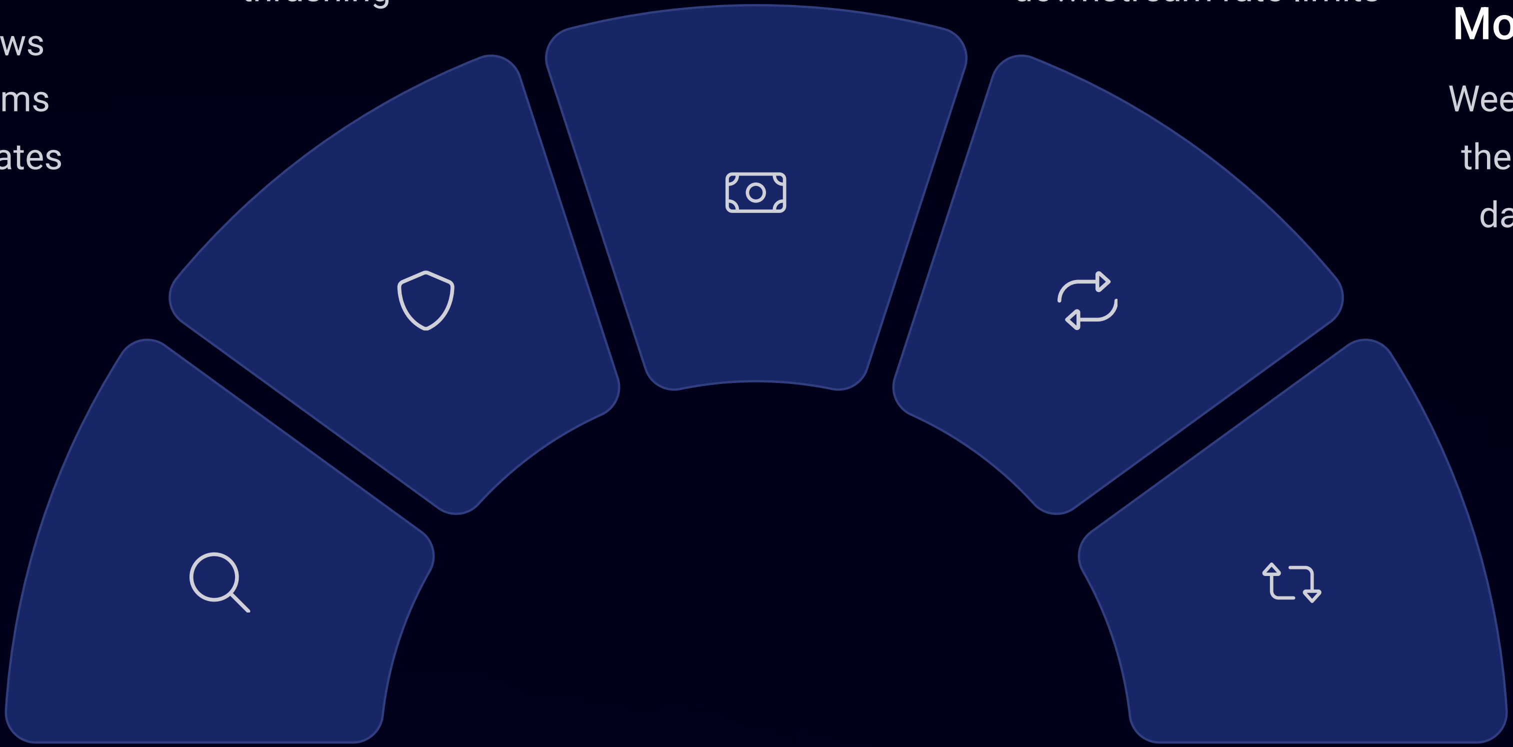
Analysis

Comparative performance across workloads, with special attention to latency distribution, cost efficiency, and error rates under different strategies.

Performance Results

Workload	Strategy	p50 Latency (s)	p95 Latency (s)	Cost (USD)	Throttle Errors
SU	Static	38.4	68.2	142.10	0
SU	Subtree	28.7	56.9	136.44	12
SU	APIx	22.3	41.8	118.79	0
SS	Static	71.6	125.2	151.02	47
SS	Subtree	55.4	96.4	148.95	19
SS	APIx	40.9	76.8	124.07	5
LE	Static	27.2	54.7	117.31	4
LE	Subtree	22.8	38.9	102.40	3
LE	APIx	21.4	32.6	99.95	1

Operational Lessons Learned



Sample Generously

Capturing every span during peak windows uncovers retry storms that low sampling rates miss

Guard-Rails Prevent Oscillation

Enforcing change caps and cool-down periods prevents system thrashing

Cost Visibility Builds Trust

Finance stakeholders embraced the AI loop once cost impacts became transparent

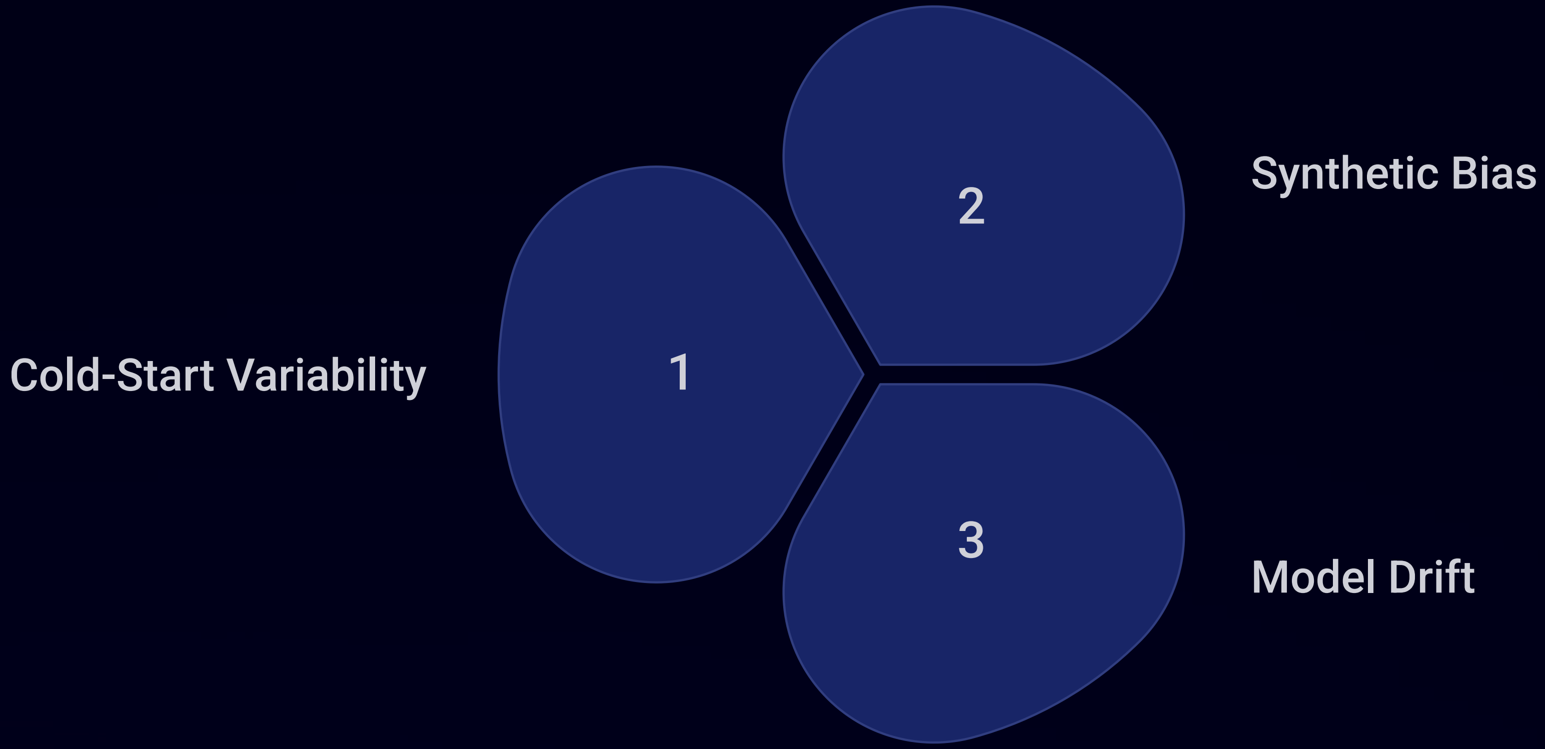
Retry Patterns Matter

High retry counts often point to payload hot-spots exceeding downstream rate limits

Model Drift is Real

Weekly retraining aligns the GNN with evolving data characteristics

Threats to Validity



1 Cold-Start Variability

2 Synthetic Bias

3 Model Drift

Conclusion and Future Work

Proven Benefits

APIx demonstrates that AI-assisted observability can evolve from passive dashboards to active control systems

Balanced Approach

Combining standard telemetry with lightweight GNN provides real-time optimization without sacrificing auditability

Future Research

Exploring cooperative tuning across multiple workflows and WebAssembly-based span feature extraction

By addressing the challenges of fine-grained telemetry capture, AI-driven insight generation, and closed-loop tuning, APIx delivers measurable improvements in latency and cost for high-fan-out serverless workloads. The framework's success demonstrates that with careful design and appropriate safeguards, AI can transform observability from a passive monitoring function into an active optimization tool.

Thank You