JACKPOT: Online Experimentation of Cloud Microservices

BY M. TOSLALI[1], S. PARTHASARATHY[2], F. OLIVEIRA[2], AND A. K. COSKUN[1]

[1]BOSTON UNIVERSITY; [2]IBM T.J. WATSON

Talk @ HotCloud
July 15, 2020

Cloud Microservices in Today's World

- Cloud microservices architecture provides agility
  - Shortens code delivery cycles
  - Enables developers to rapidly innovate
- Agile practices encapsulate:
  - Continuous deployment
  - **Online experimentation**

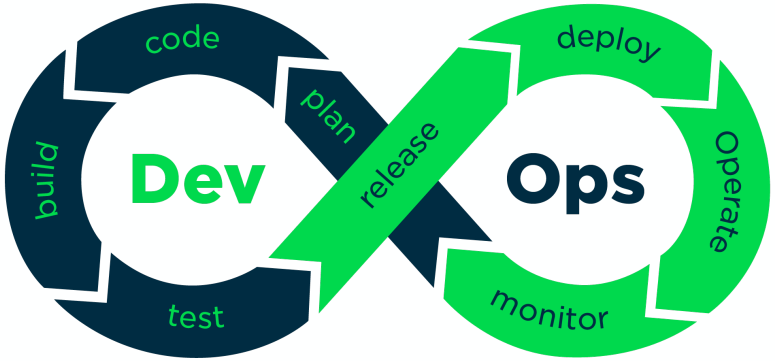


Figure from cisco.com

Web & Mobile Online Experimentation

- Goal: Compare multiple versions of a component in production to identify “best” one
- Versions are subject to single KPI[1] (reward, e.g., CTR[2])

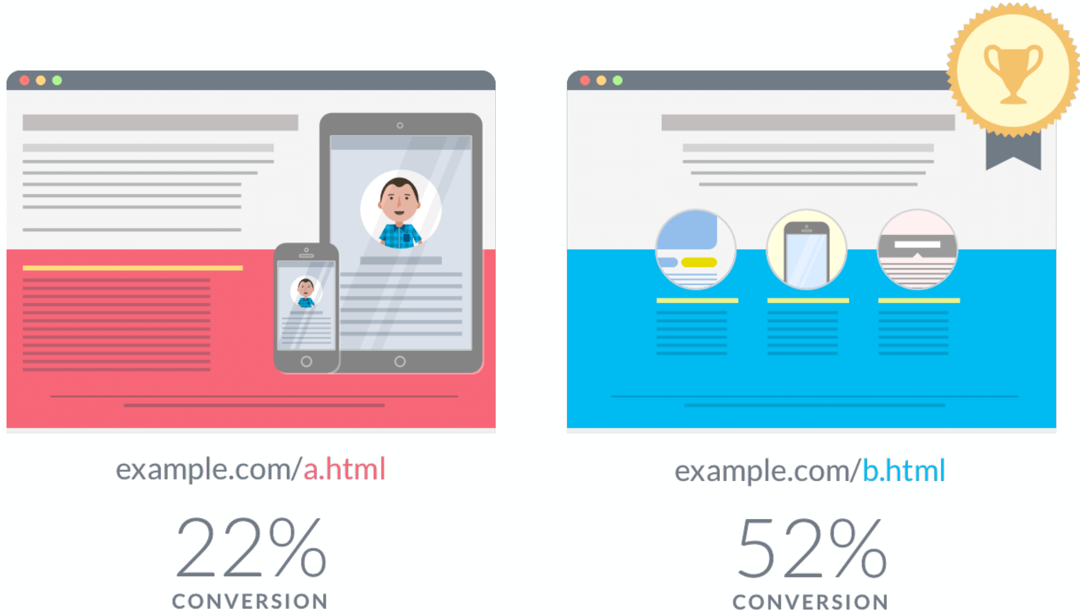


Figure from optimizely.com

[1] Key performance indicator
[2] Click-through rate

Cloud Challenges

- Cloud is volatile due to:
  - Resource contention
  - Failures
  - Latency
- Profound financial and reputation damages

- **Necessity:** multi KPI experiments
  - Latency along with a reward

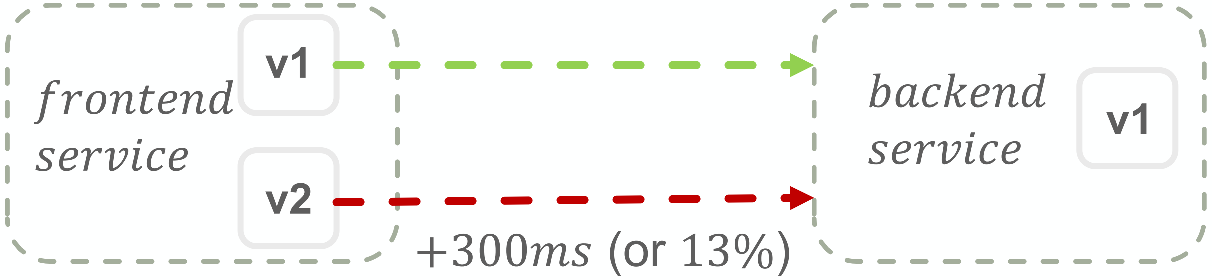
Half a second delay caused a 20% drop in traffic[3]

Every $100ms$ of latency cost 1% in sales[4]

[3] http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html
[4] https://www.gigaspaces.com/blog/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/

Further Challenges Posed by Microservices

- Interactions between microservices can affect the overall user-perceived performance and correctness
- Canopy [Kaldor et al., 2017] describes a scenario on Facebook.com



- **Necessity:** Experiment with combination of microservices (i.e., path)
  - E.g., path = $frontend\_v2, backend\_v1$

Jackpot: *Online Experimentation of Cloud Microservices*

- We propose a novel formulation for online experimentation of cloud microservices
  - Generalizes traditional approaches used in mobile & web environment
  - Encapsulates challenges posed by the cloud environment
- To enable developers to apply our formulation:
  - We present the system *"Jackpot: Online Experimentation of Cloud Microservices"*

Design Choices

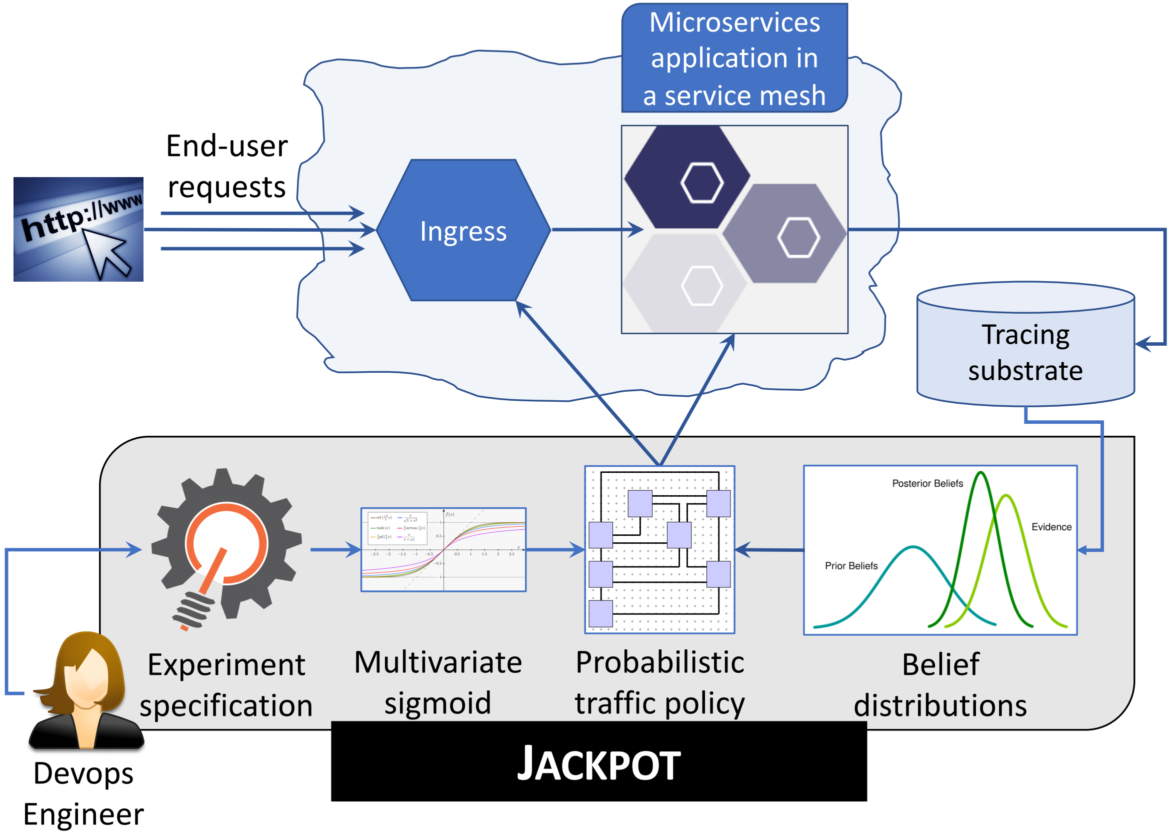
1) **Multivariate experiments**
   - Identify the best *path* instead of best version on a single service
2) **Multi-KPI experiments**
   - Express preferences in an experiment using multiple KPIs (e.g., CTR + latency)
   - Hard and soft constraints on KPIs
3) **Multi-types of experimentation**
   - Best path identification
   - Utility maximization
   - Pure statistical estimation

Jackpot Internals

Istio service mesh provides:

1) **Traffic management**: Mesh should be dynamically configured to issue traffic split between paths

2) **Distributed tracing**: Ability to assess and compare a combination of microservices

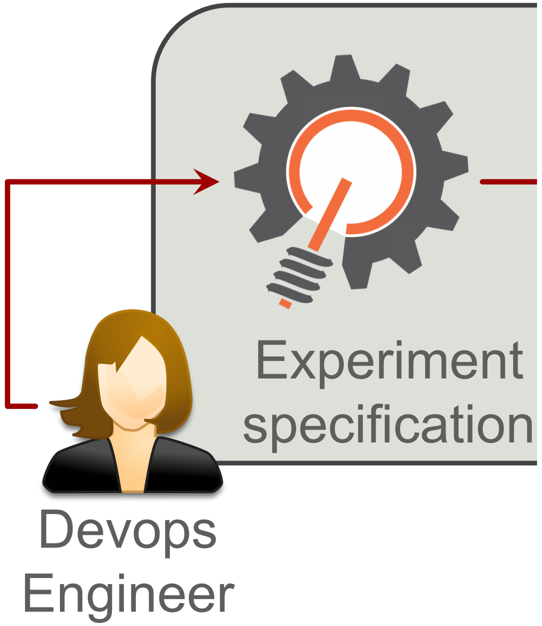


Jackpot injects headers to incoming requests in the course of an experiment:

1) Enables traffic routing according to a *path*
2) Collects *path* specific KPIs

Jackpot’s Workflow

```yaml
services:
  - reviews
  - ratings
  - details
KPIs:
  latency:
    limit: 500ms
  ctr:
    reward: True
```

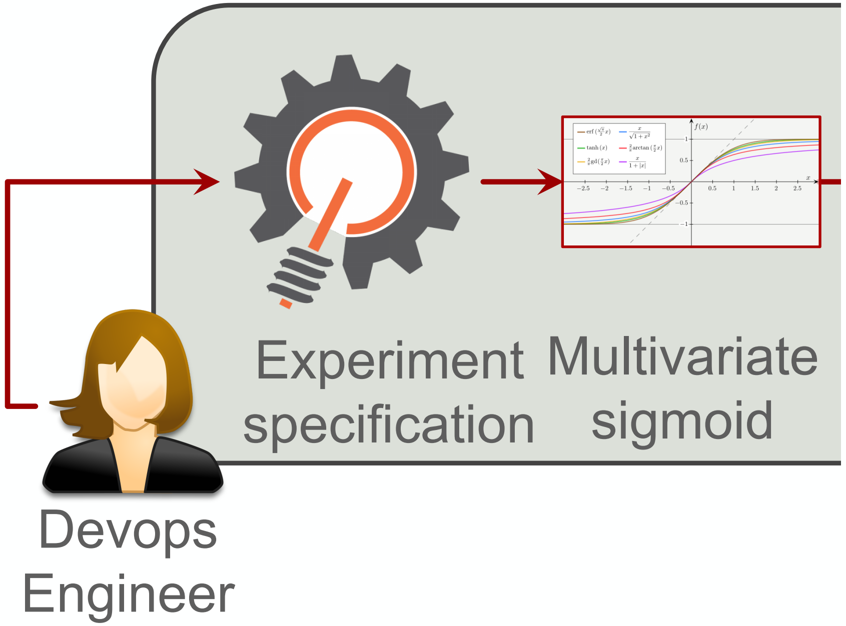


Jackpot input: *Experiment Spec*

- Provided as a YAML file
- Contains:
  - Services
  - KPIs

Multivariate Sigmoid

```
services:
  - reviews
  - ratings
  - details
KPIs:
  latency:
    limit: 500ms
  ctr:
    reward: True
```

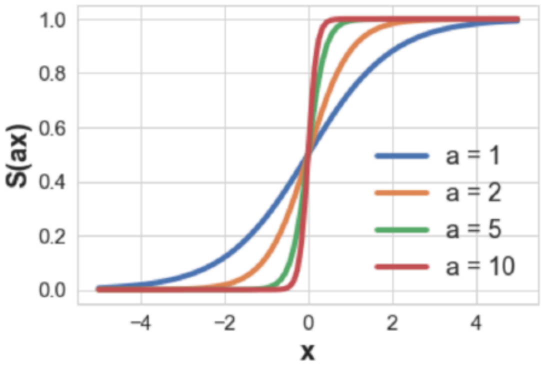


Utility

$$h_a(p) = \mathbb{E}[X_0[p]]\Pi_{j=1}^{k} S\left(a\left(1 - \frac{\mathbb{E}[X_j[p]]}{\ell_j}\right)\right)$$

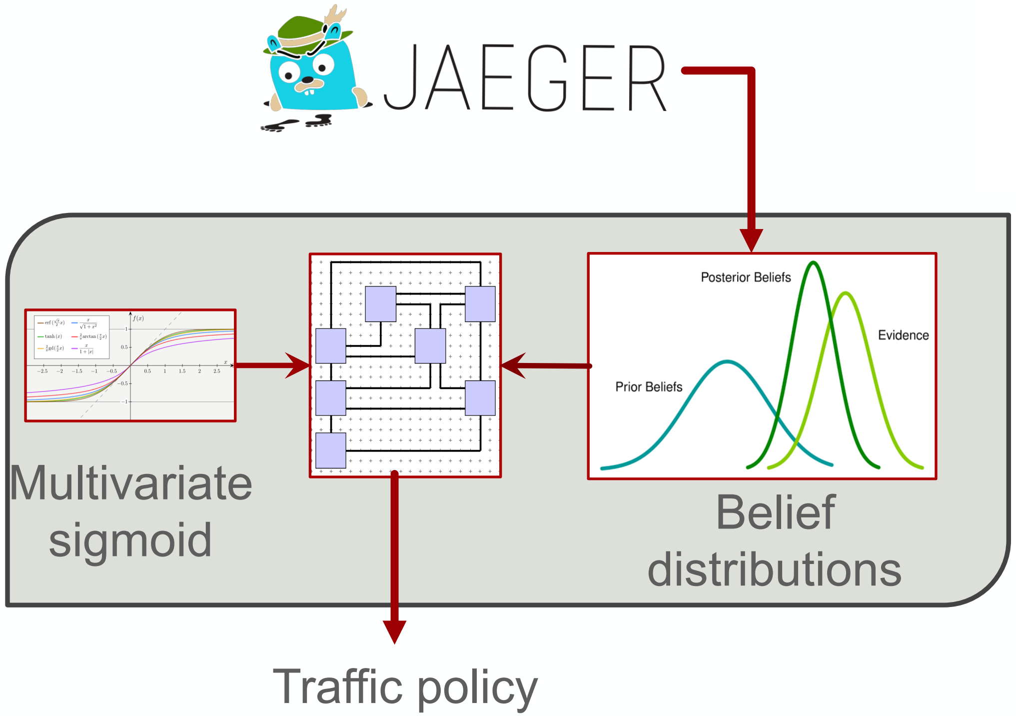
a: Amplification, X_j: KPI, ℓ_j: Constraint

1) *Combine multiple KPIs into one*
2) *Flexibility: Hard & Soft constraints*



Scaled sigmoid

Online Learning



Utility

$$h_a(p) = \mathbb{E}[X_0[p]]\Pi_{j=1}^{k} S\left(a\left(1 - \frac{\mathbb{E}[X_j[p]]}{\ell_j}\right)\right)$$

Utility components need to be learned online

Jackpot maintains Bayesian belief distributions

Monte Carlo sampling answers:

1. What is the estimated utility of path *p*?
2. What is the probability of p being optimal?

Holistic Algorithm: Top-k Sigmoid Thompson Sampling

- Thompson Sampling (TS) is a provably robust multi-armed bandit algorithm
- Multi-armed bandit: exploration vs. exploitation dilemma
- k-STS samples from belief distributions and plug these into the *sigmoid* function (Monte Carlo)
  - Finally chooses top-k paths uniformly at random

1-STS

- Generalized version of TS
- Exploits the best path
  - Type1: Utility maximization

2-STS

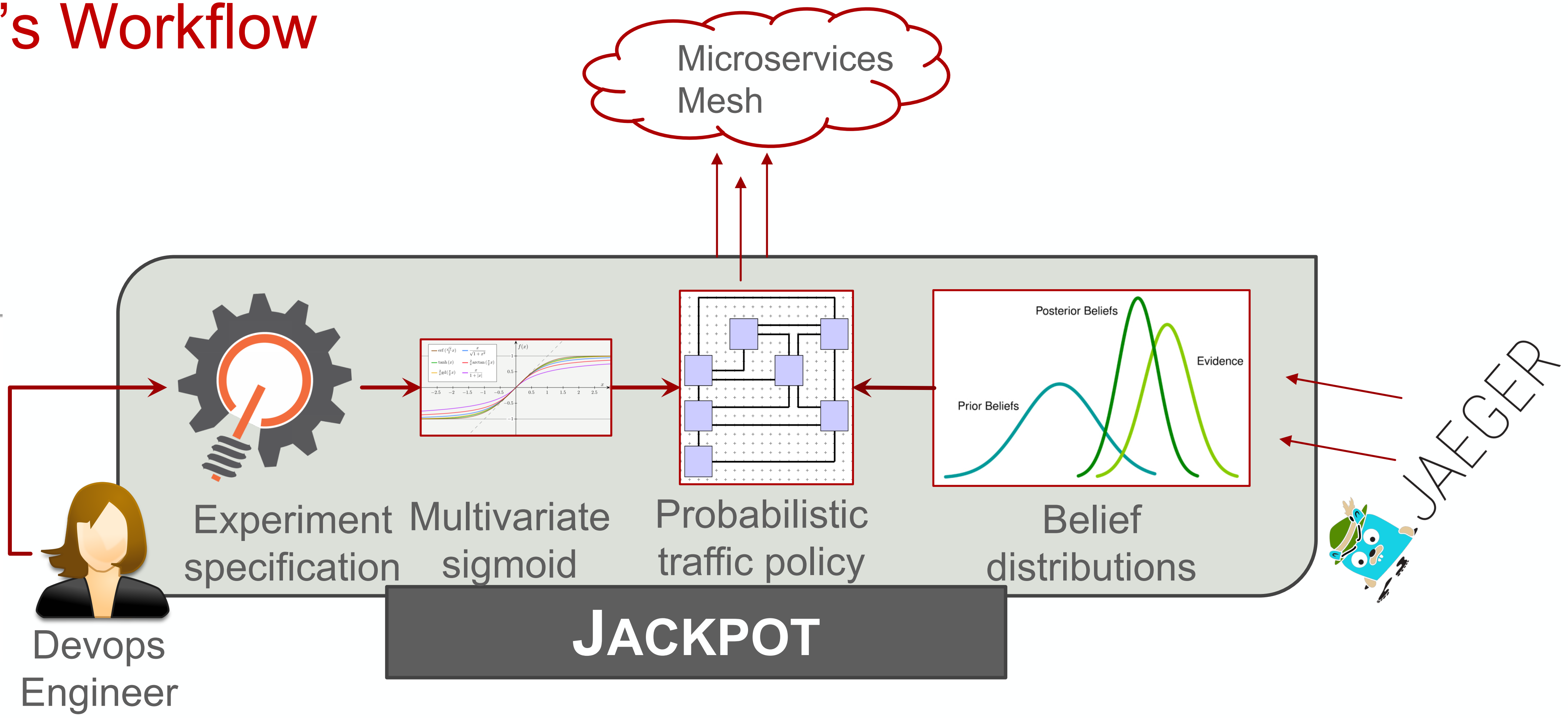
- Generalized version of Top-two TS
- Explores the best and an alternative
  - Type2: Best path identification

N-STS/UNIF

- Uniform policy (***UNIF***)
- Evaluates each candidate equally
  - Type3: Statistical estimates

Jackpot’s Workflow

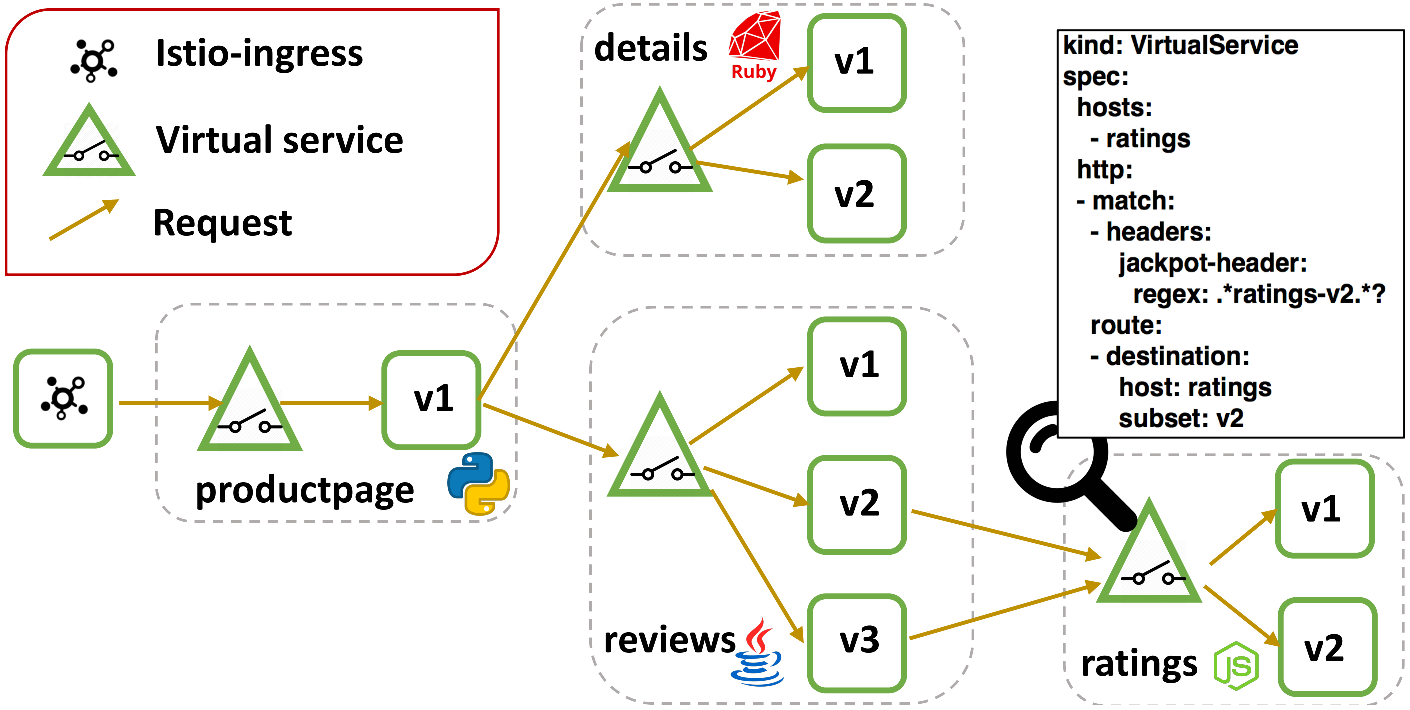
```
services:
  - reviews
  - ratings
  - details
KPIs:
  latency:
    limit: 500ms
  ctr:
    reward: True
```



Experiments

We evaluate the performance of 1-STS, 2-STS, UNIF.

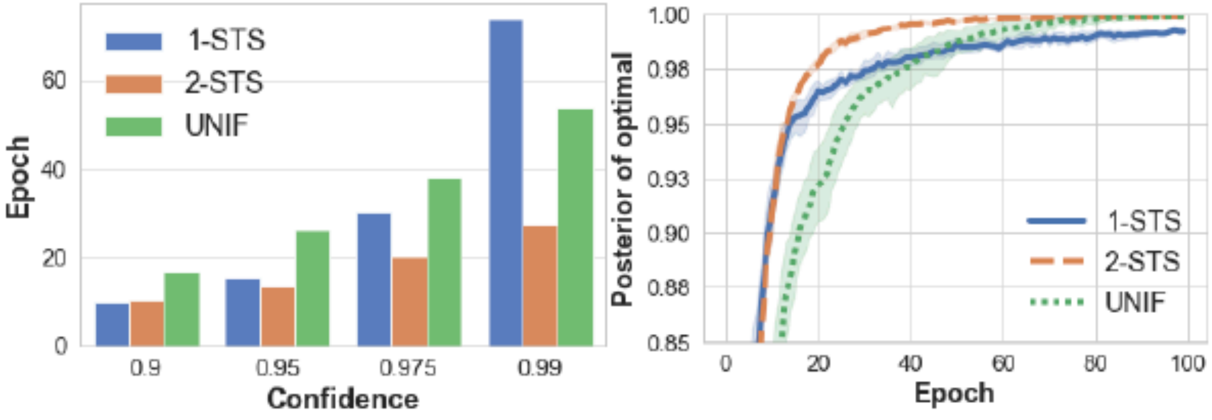
- Constraint on mean latency
  - i.e., $E[X1[p]] \;<=\; 300ms$
- Set $a = 10$ ➡ hard constraint
- Workload: $50\,reqs/epoch$
- $100\;epochs, \;\; 5\,runs$



Bookinfo application

Best Path Identification

- 1-STS struggles to reach higher confidence levels
  - Selects the optimal in almost all periods
- 2-STS prevents focusing on one candidate
  - Top-2, the best or an alternative is chosen



# of epochs per level   Posterior per epoch

Best path identification experiment

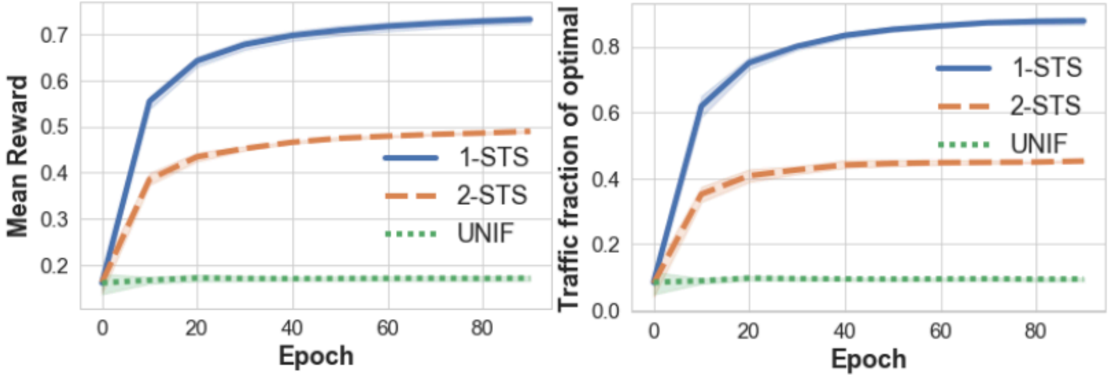
- 2-STS requires 49% fewer epochs compared to UNIF, and 63% fewer compared to 1-STS

Utility Maximization

Experiment setting.

#	path	$\mathbf{X}_1$ (ms)	$\mathbf{X}_0$	$h_a(p)$
7*	$pp_{v1}, det_{v2}, rev_{v1}$	253	0.77	0.64

- Observe that 1-STS maximizes the reward during experimentation
  - True reward of optimal = 0.77
- 1-STS works toward exploiting the optimal, thus maximizing the utility



Mean Reward Accuracy

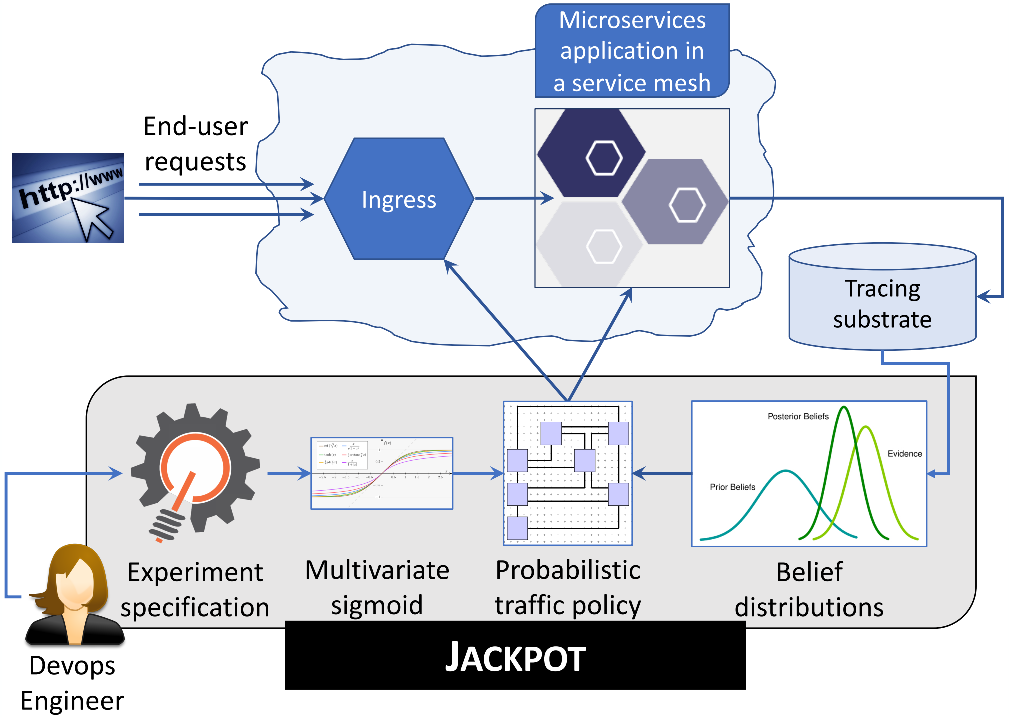
Reward maximization experiment.

Next Steps

- Dynamic incorporation of versions as they arrive into ongoing experiments
- The ability to handle heterogeneous cloud applications
  - Absence of header propagation → No path-level traffic splitting
  - Absence of distributed tracing → Multi-type telemetry functionality

THANK YOU

Contact: toslali@bu.edu



Jackpot: *Online Experimentation of Cloud Microservices*

- Online experimentation on a combination of microservices (i.e., paths)
- Multi-KPI experiments
- Multi-types of experimentation