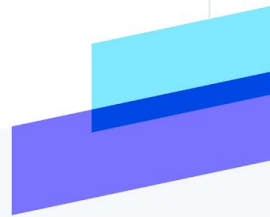


stripe

Health Checks: A Boon or a Curse?



Who Are We?



John Murray

Service Networking Engineer, Stripe
Occasional Envoy Contributor
C++ Enthusiast



Venil Noronha

Service Networking Engineer, Stripe
Envoy and Istio Contributor
Distributed Systems Enthusiast

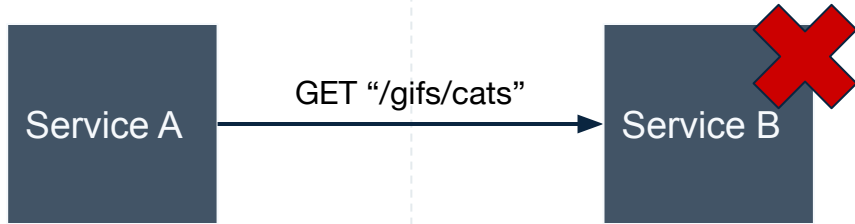
Agenda

- 1 Conceptual Overview
- 2 Options within Envoy
- 3 Health Checking at Stripe
- 4 Problems at Scale
- 5 Conclusion

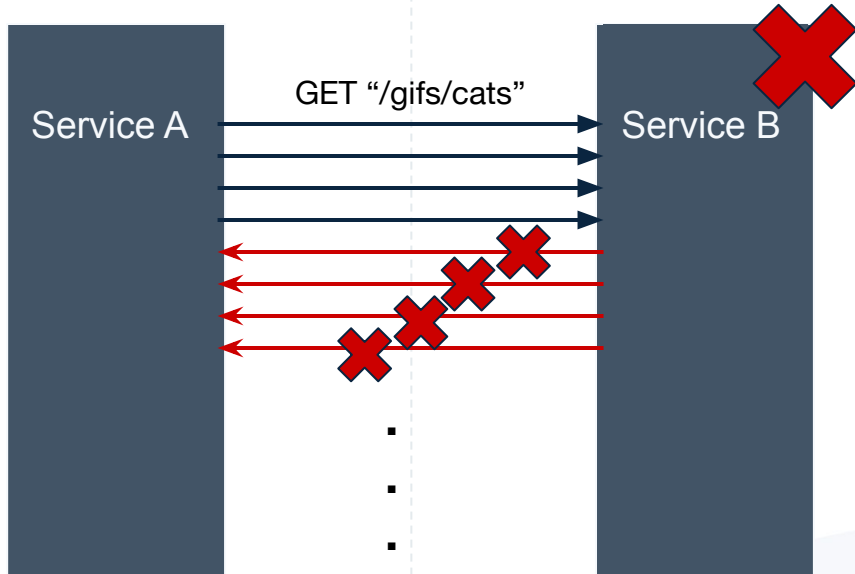
Agenda

- 1 **Conceptual Overview**
- 2 Options within Envoy
- 3 Health Checking at Stripe
- 4 Problems at Scale
- 5 Conclusion

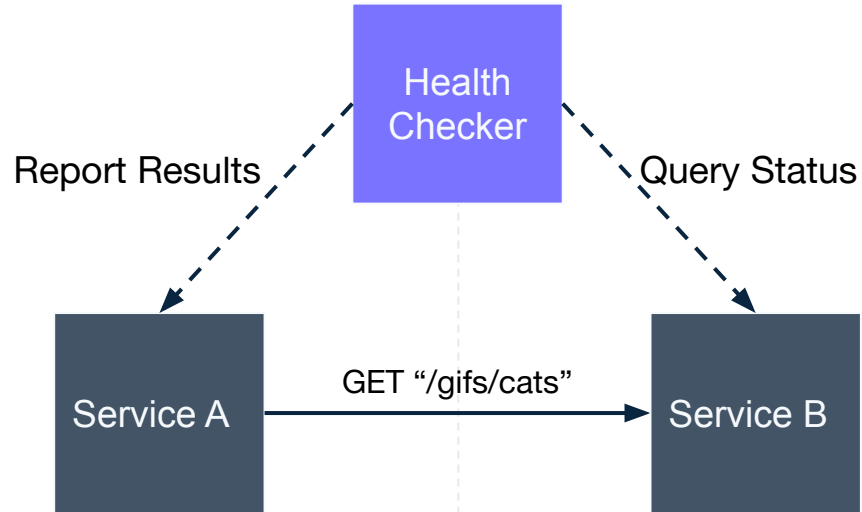
Health Checking



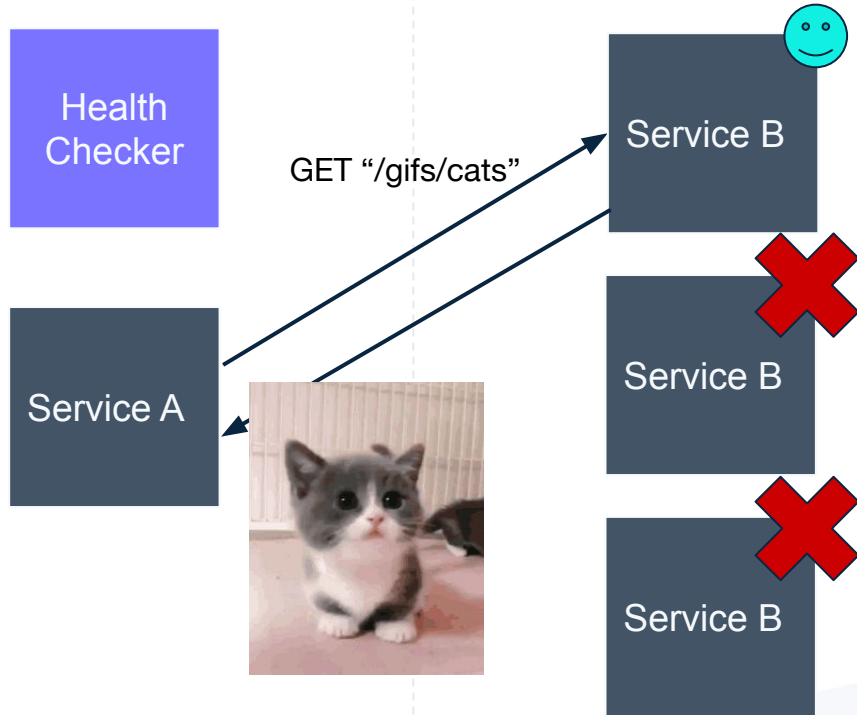
Health Checking



Health Checking



Health Checking



Agenda

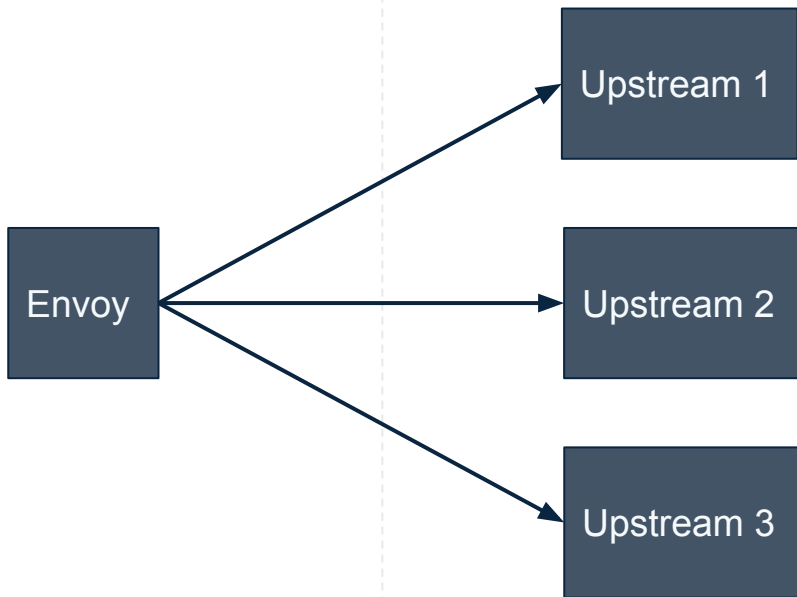
- 1 Conceptual Overview
- 2 **Options within Envoy**
- 3 Health Checking at Stripe
- 4 Problems at Scale
- 5 Conclusion

Options

1. Active Health Checks
2. Passive Health Checks (Outlier Detection)
3. Externally Sourced Health

Active Health Checks

Configure Envoy to explicitly health check upstreams

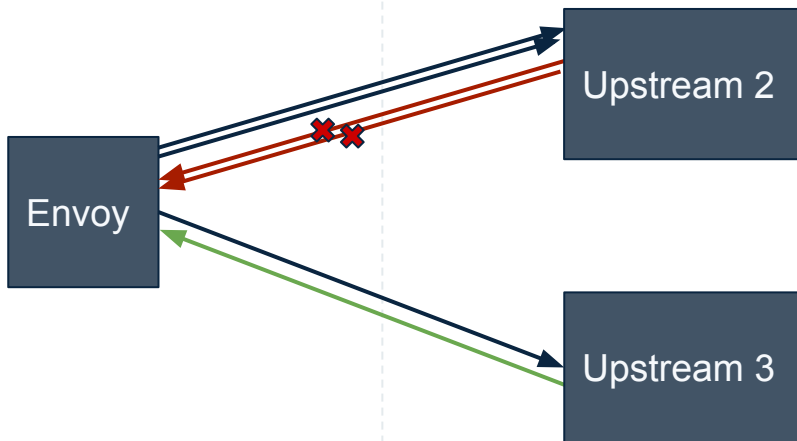


```
cluster:  
  health_checks:  
  - interval: 5s  
    timeout: 4s  
  http_health_check:  
    host: 'mycluster.service.envoy'  
    path: '/healthcheck'
```

Passive Health Checks

Also known as “Outlier Detection”

Examines traffic properties such as response-codes and latency.



```
# Detect continuous 5xx errors
```

```
cluster:
```

```
  outlier_detection:
```

```
    consecutive_5xx: 20
```

```
    max_ejection_time: 120s
```

```
# Detect timeout / connection issues
```

```
cluster:
```

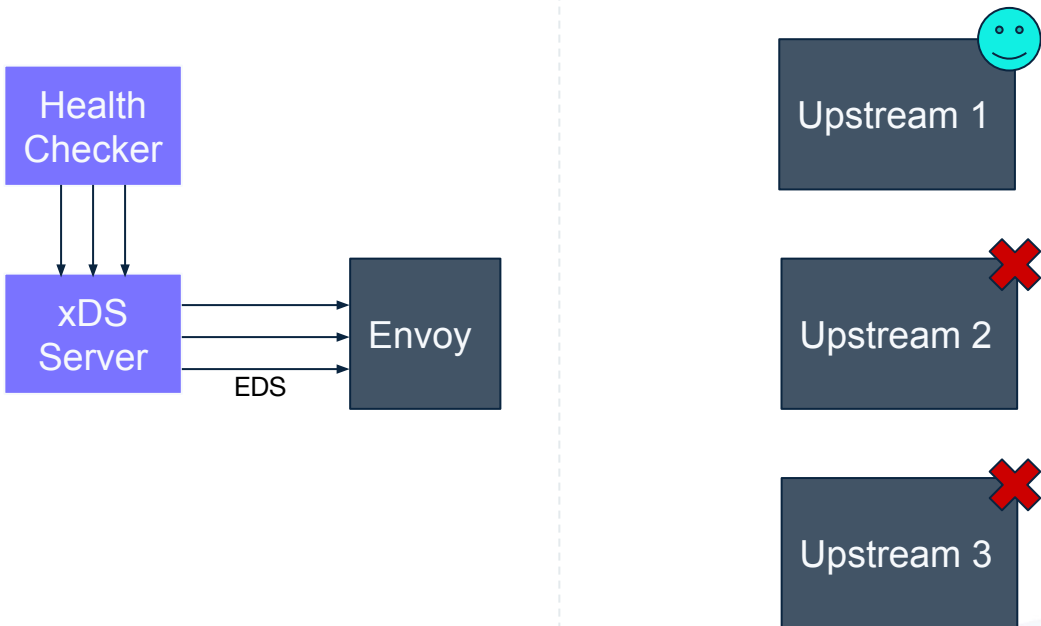
```
  outlier_detection:
```

```
    split_external_local_origin_errors: true
```

```
    consecutive_local_origin_failure: 5
```

Externally Sourced Health

Health checked by an external system, provided via xDS.



```
lb_endpoints:  
- endpoint: '{...}'  
  health_status: 'HEALTHY'  
  
- endpoint: '{...}'  
  health_status: 'UNHEALTHY'  
  
- endpoint: '{...}'  
  health_status: 'DRAINING'
```

Agenda

- 1 Conceptual Overview
- 2 Options within Envoy
- 3 **Health Checking at Stripe**
- 4 Problems at Scale
- 5 Conclusion

Health Checking at Stripe

- Active Health Checks via Envoy
 - Cached via proxied internal-cluster
 - Aggregates “host health” with service health
 - Local, active draining

Historically has worked well:

- Low-volume, latency sensitive connections
- Traffic shifting controls

Agenda

- 1 Conceptual Overview
- 2 Options within Envoy
- 3 Health Checking at Stripe
- 4 **Problems at Scale**
- 5 Conclusion

Problems

- Slow time-to-detection
 - Problematic for high-volume connections
- High Health Check Volume
 - Shared (foundational) services have many downstreams
 - Active health probes can add significant overhead
- Network Costs
 - Health Checks will traverse all network paths regardless of endpoint weights or LB strategies

Problems: Slow Time-To-Detection

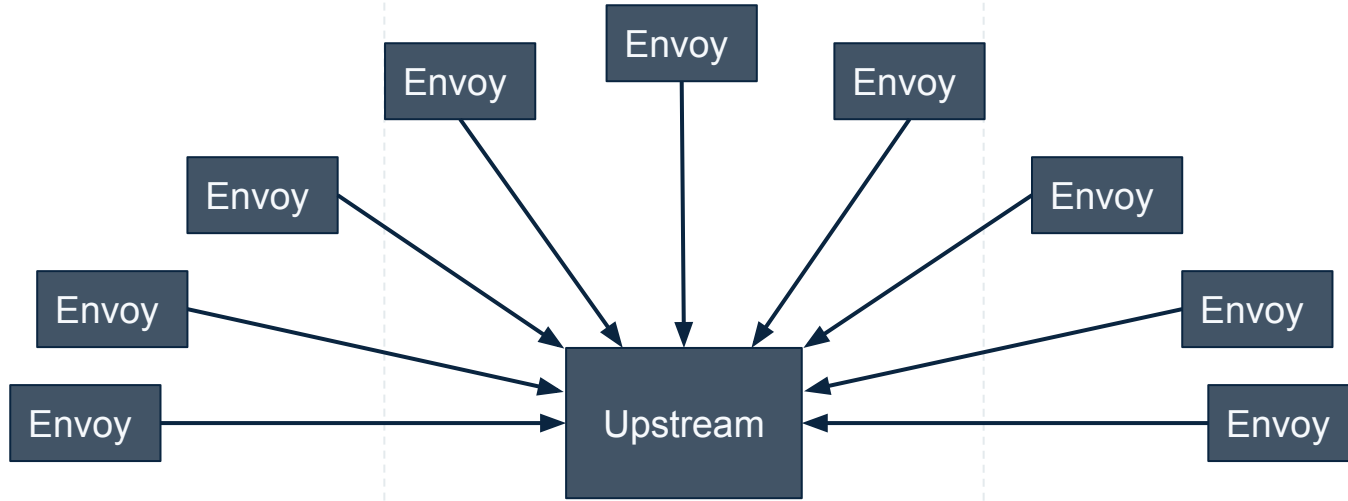
Example

- Active health check interval of 5s
- Network link processes 25K qps
- Potential impact: $5 * 25K = 125K$ failed requests

Mitigations

- Reduce the active health check interval
 - Additional steady-state costs
- Mitigated by passive health checking (outlier detection)
 - Faster than active health checking
 - Lower overhead compared to active health checking
 - Configured in combination with active health checking

Problems: High Health Check Volume



Problems: High Health Check Volume

Issues Encountered

- High backend load
- High connection load
- High CPU load

Anecdotes

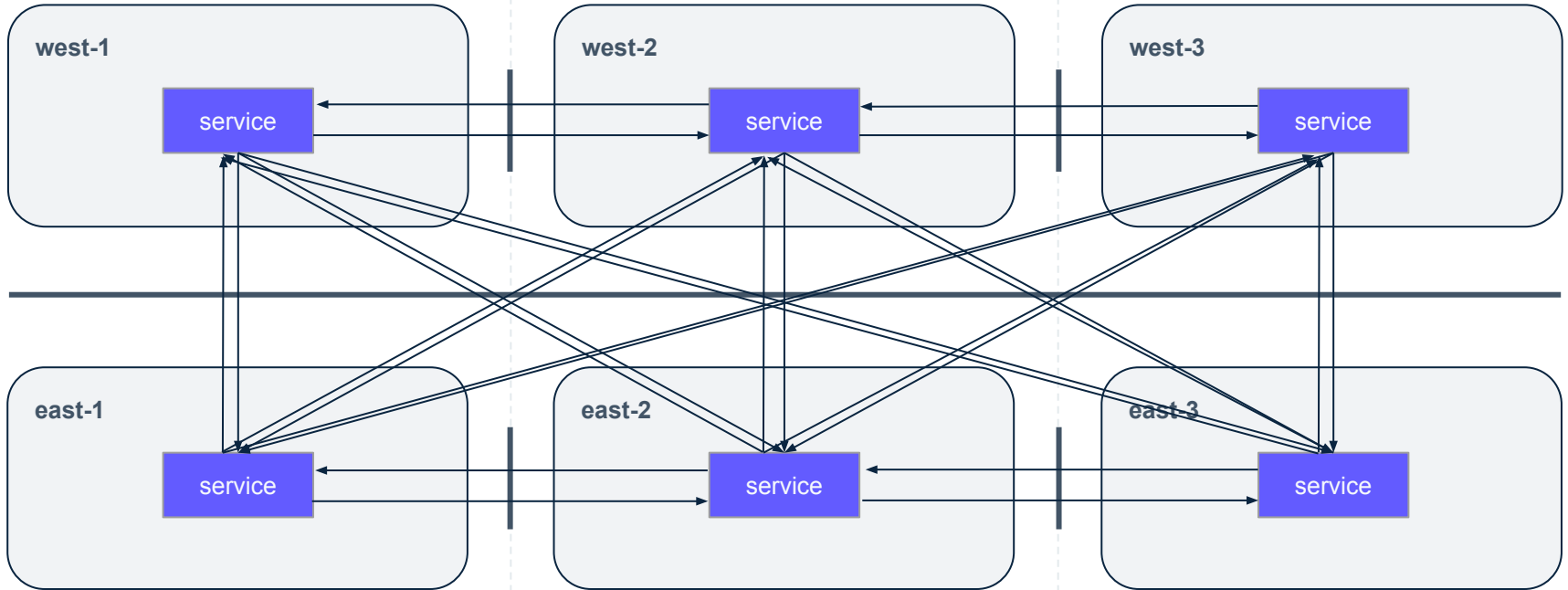
- Burning 2+ CPU cores exclusively to serving health check traffic
- Deploying larger instance types to account for volume
- Observed general slowdown of Envoy
- Reaching host-defined file descriptor limits

Problems: High Health Check Volume

Mitigations

- Cache backend health*
- Reduce health check interval*
- Reduced reachability
- Control-plane subsets*
- Move to centralized health checks*

Problems: Network Costs



Problems: Network Costs

Mitigations

- Increased health check interval*
- Exclusively use passive health checks*
- Reduced reachability
- Control plane subsets*
- Move to centralized health checks*

Agenda

- 1 Conceptual Overview
- 2 Options within Envoy
- 3 Health Checking at Stripe
- 4 Problems at Scale
- 5 **Conclusion**

Conclusion

- All strategies are context-dependent
- Just one strategy is maybe not enough
- The data-path is never 100% safe
- Let scale drive your design

stripe

Thank You!

We're Hiring!

John Murray | murray@stripe.com

Venil Noronha | venil@stripe.com