envoycon
NORTH AMERICA

SLATE

Service Layer Traffic Engineering

*Gangmuk Lim
PhD student
UIUC

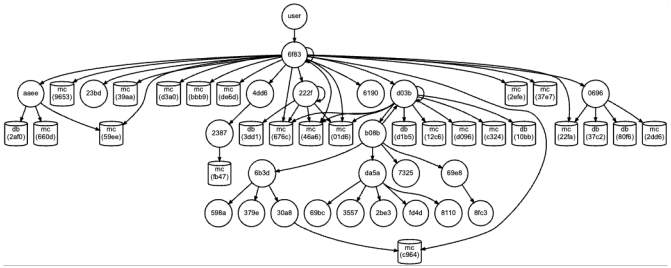
*Aditya Prerepa
Undergrad
UIUC & Aviatrix

Brighten Godfrey
Professor
UIUC & VMware

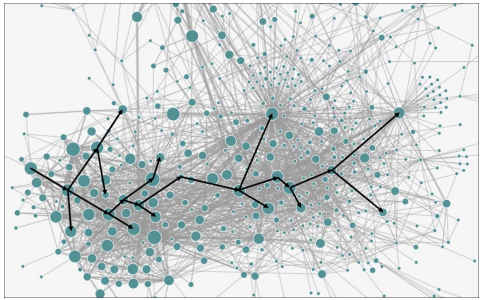
Radhika Mittal
Assistant Professor
UIUC

Microservice applications are complex!

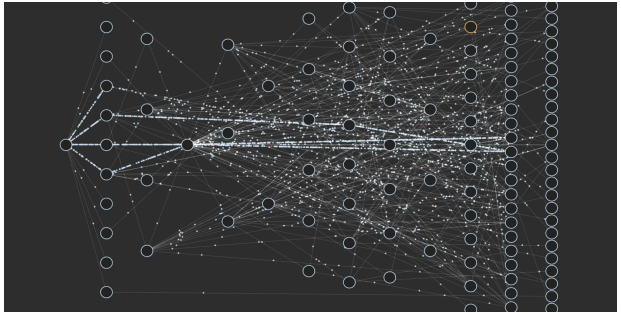
Optimizing complex microservice application is challenging



Reference: Alibaba microservice call trace

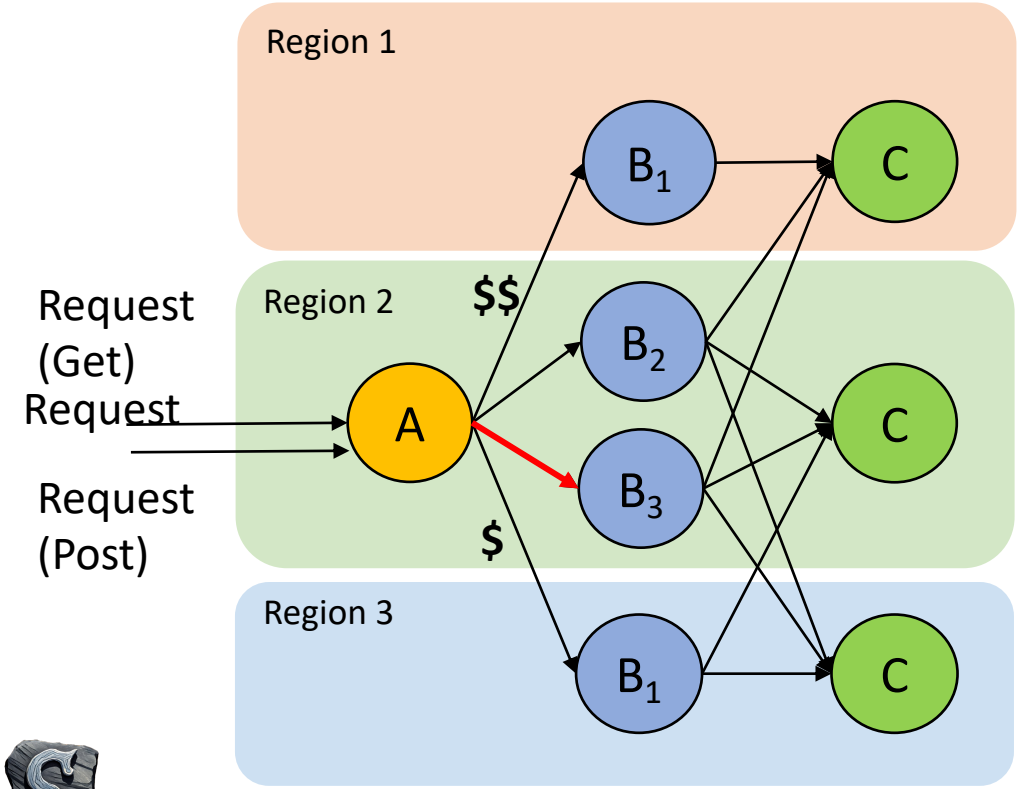


Reference: Uber microservice visualization



Reference: Netflix microservice visualization

What could be a good request routing?



Network RTTs

- Geo-distributed regions (Region1,2,3)
- Locality within the region (B2, B3)
- Network congestion

Cost

- Bandwidth (egress cost)
- Compute (VM cost)

Different requests

- Different resource implication
- Different call graph

Multi-hop implication

Survey: Istio users' cross-cluster routing today

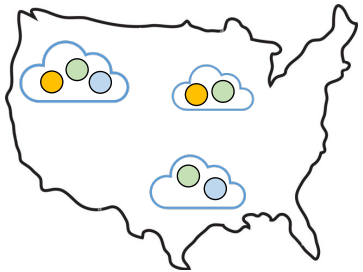
The survey will be publicly available in UIUC ***ServiceLayerNetwork*** website!

Multi-cluster services => Optimal routing nontrivial

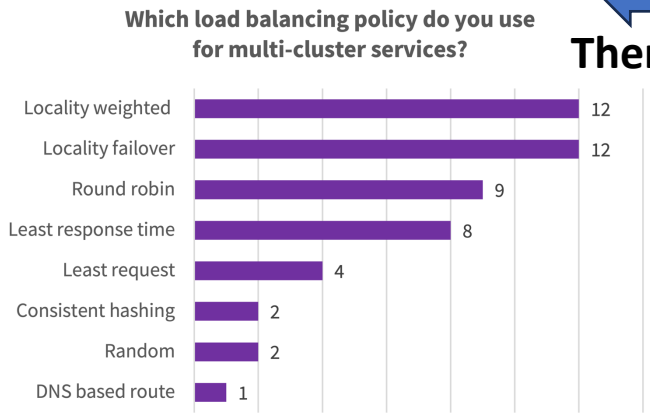
Respondents are still relying on relatively simplistic load balancers.

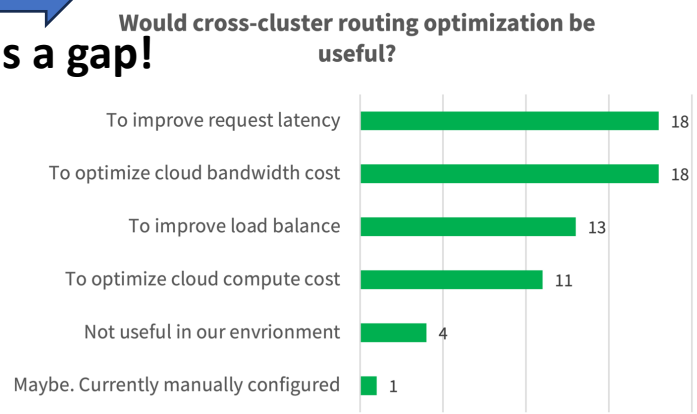
They want to optimize routing to minimize latency and cost.

Up to 50 or more clusters
50% multi-cluster services



There is a gap!





Optimal request routing!

- **Current approach:**

  Request routing is much more subtle problem than today's load balancers and traffic management

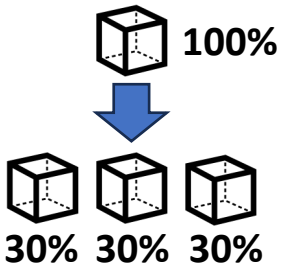
- **Goal:**

  Globally optimal request routing based on administrator intent.

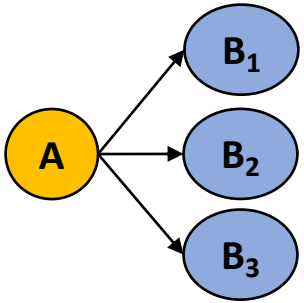
I thought this problem was solved!?

*We already have fancy **autoscaler, load balancer, and traffic management !?***

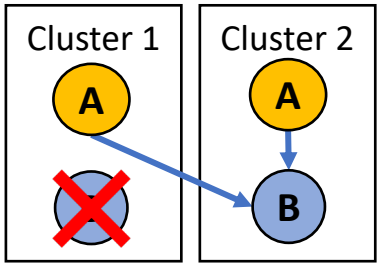
Autoscaler



Load balancers



Traffic management



*locality failover

Advanced traffic management

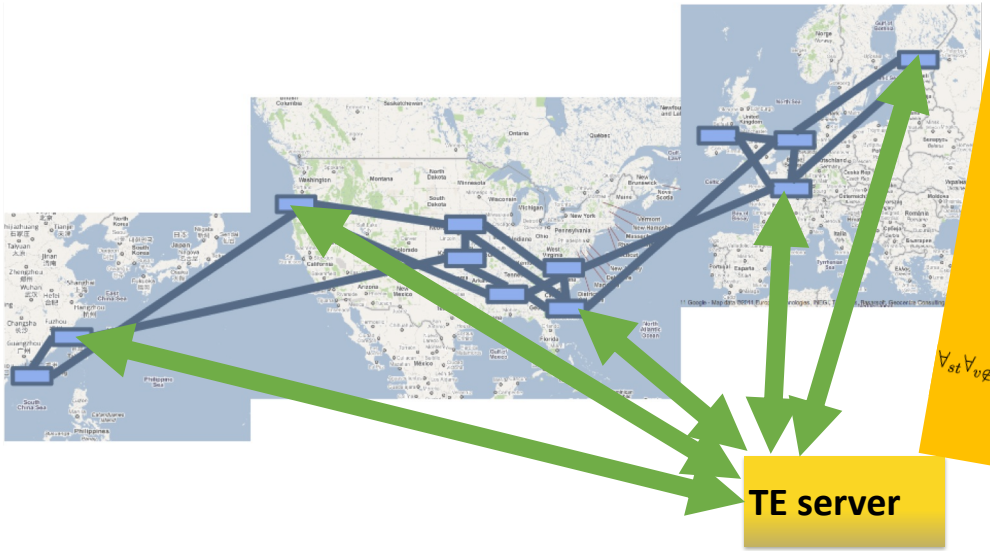
It does not consider

- network RTTs (advanced traffic management does)
- network costs
- different requests
- multi-hop implication

*Advanced traffic management: Google's Traffic Director, Meta's ServiceRouter [Saokar et al, OSDI '23]

SLATE: Service Layer Traffic Engineering

Optimizing global request routing becomes a traffic engineering problem!



Given

C_{ij} Capacity of link from i to j

T_{st} Traffic demand from s to t

Objective

$\min u^*$ Minimize max link utilization

Subject to

$0 \le u^* \le 1$

$x_{ijst} > 0$

$\forall_{st}\ T_{st} = \sum_j x_{sjst}$ Traffic volume of (s,t) flow carried over link (i,j) is > 0

$\forall_{st}\ T_{st} = \sum_i x_{itst}$ Ingress flow equals demand

$\forall_{st} \forall_{v \notin \{s,t\}} \sum_i x_{ivst} = \sum_j x_{vjst}$ Egress flow equals demand

$\forall_{i,j} \sum_{s,t} x_{ijst} \le u^* C_{ij}$ Flow conservation

Utilization cap

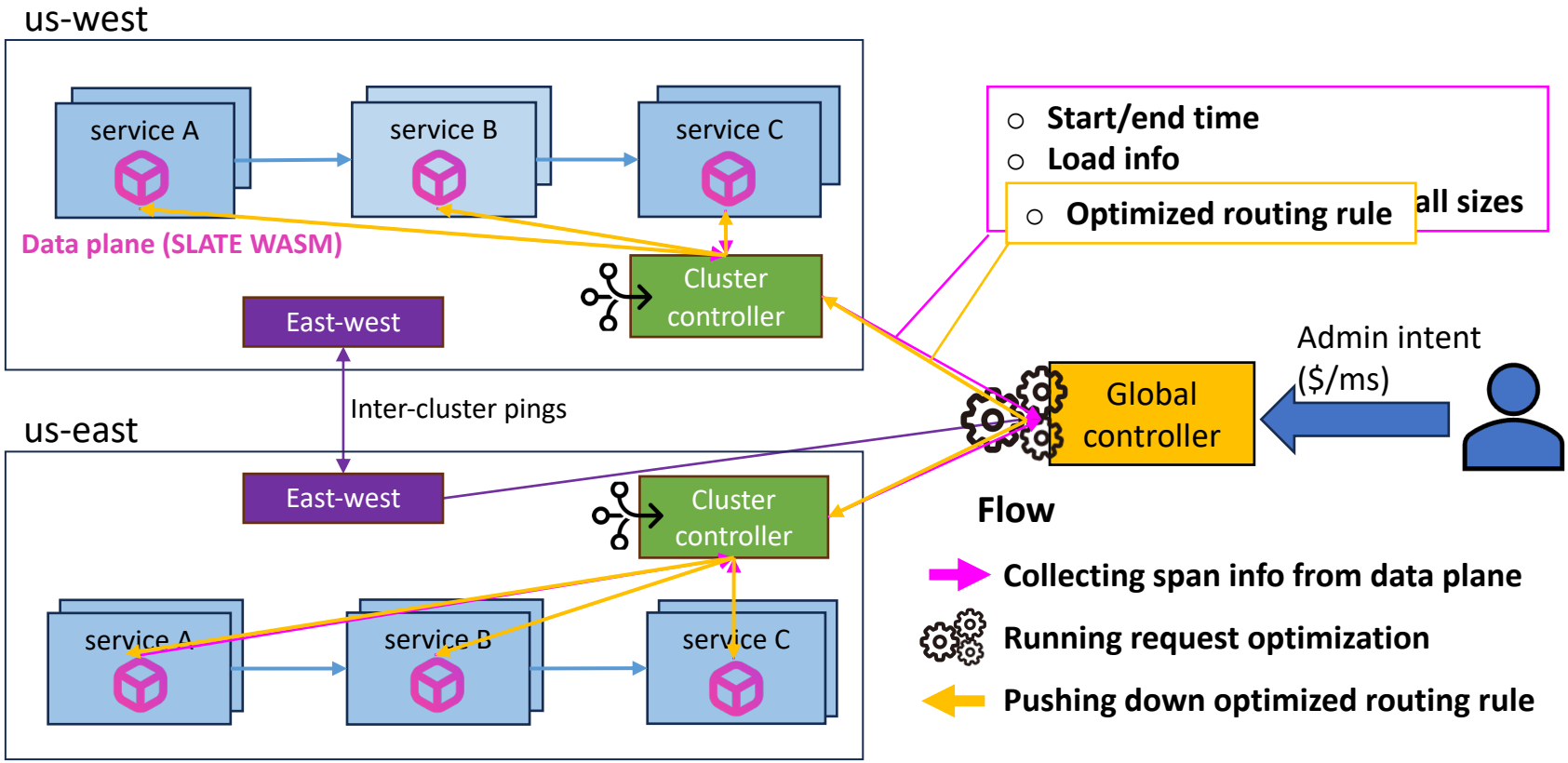
E.g., Microsoft's SWAN [Hong et al, SIGCOMM'13], Google's B4 [Jain et al, SIGCOMM'13]

Design Goals of SLATE

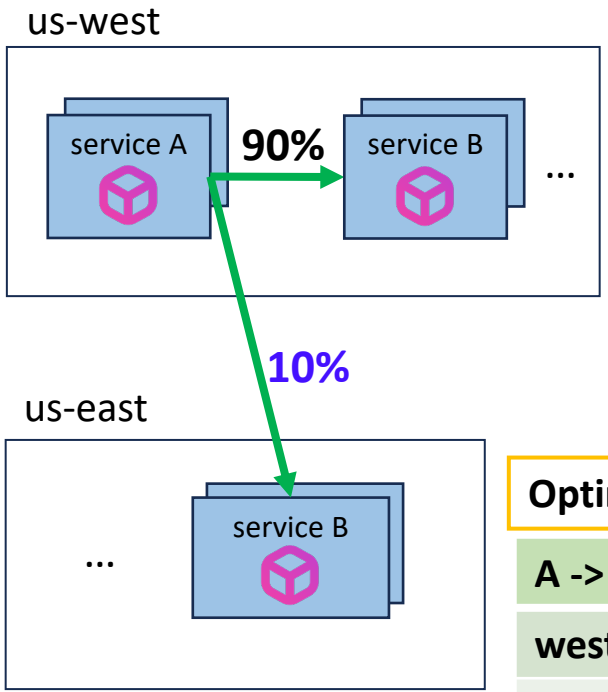
Automatically optimizing global request routing

- Fast reaction to changes in load and latency
- Ease of expression for multidimensional operator intent
  - Tradeoffs between cost/latency/availability
- Instant pluggability into existing deployments
  - service mesh, fleet of load balancers, RPC library.

SLATE Architecture



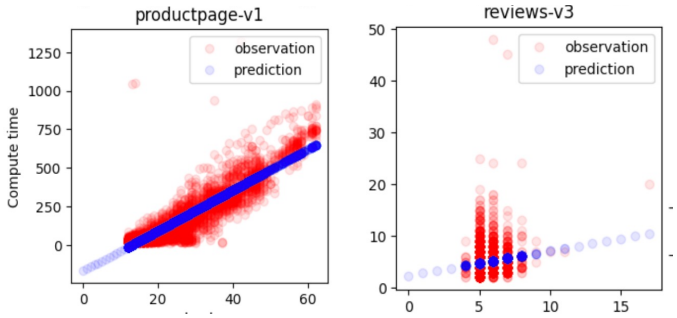
SLATE Global controller

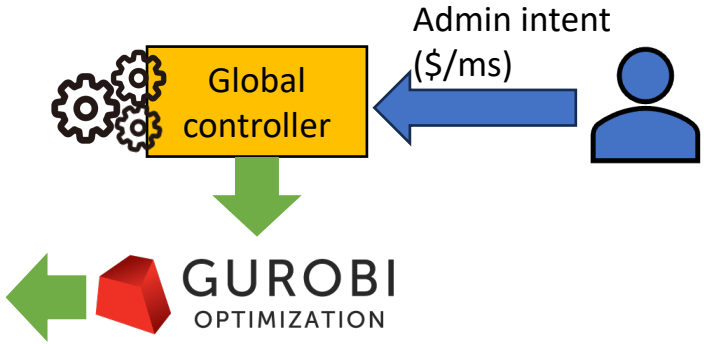


Optimized routing rule

A -> B	west	east
west	90%	10%
east	0%	100%

Modeling latency as a function of load





Linear programming

Use Cases

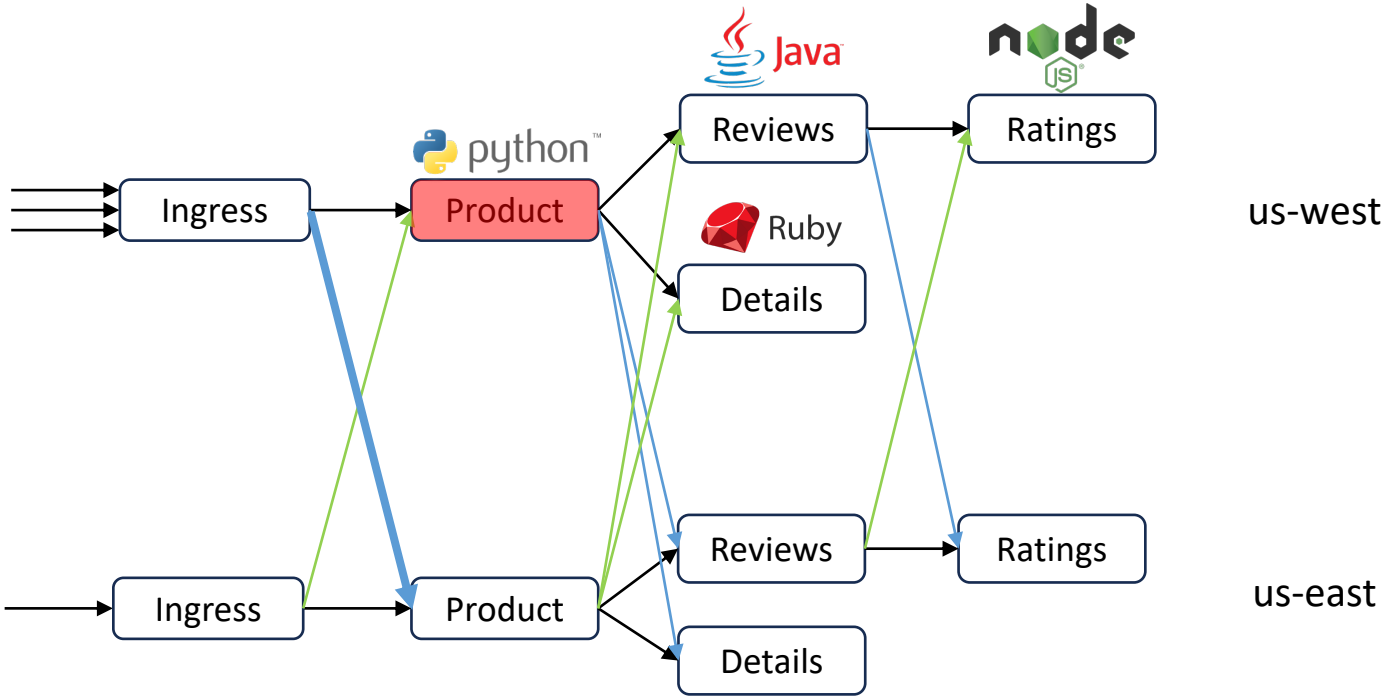
Use case 1: Instantly reacting to fluctuations in load.

Use case 2: Optimizing Egress Cost in dynamic microservice topologies.

Use case 3: Classifying and handling different call graphs and request types.

Use case 1: Microburst in cluster 1

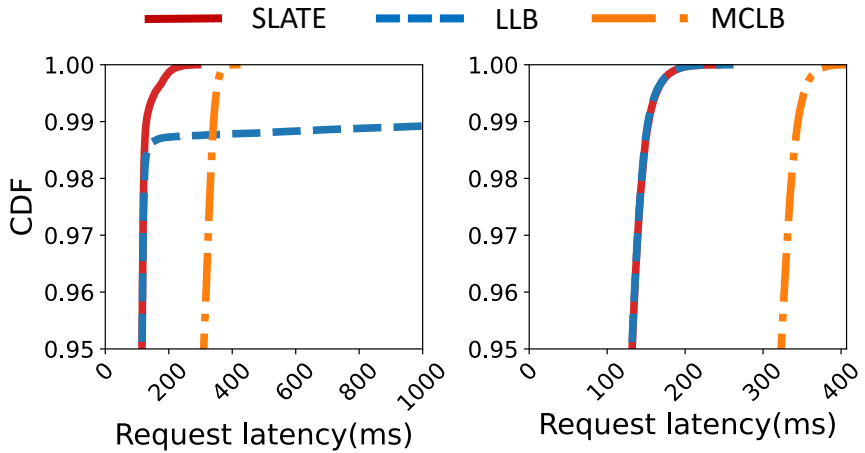
How should we distribute requests across replicas in different regions?



*Istio bookinfo application

Use case 1: Microburst in cluster 1

- **Local Load Balancing (LLB)**: Route requests to instances **within the local cluster**.
- **Multi-Cluster Load Balancing (MCLB)**: Route requests to instances **across all clusters**.
- **SLATE**: Route requests to the remote cluster if the local cluster does not have an underutilized replica and the remote cluster has one. Otherwise, route to the local cluster.



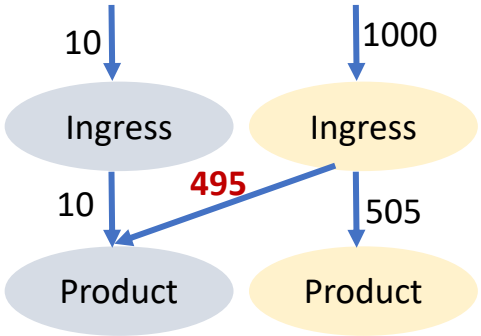
$/ms in use case 1

What's the value of the latency in your case?

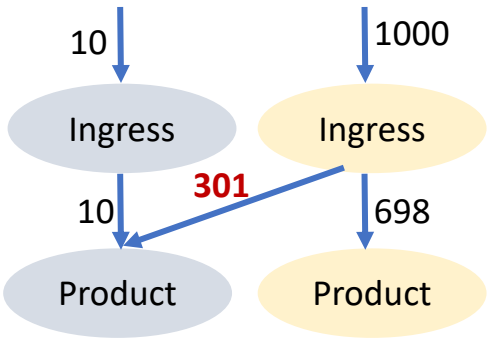
Latency is more important

Cost is more important

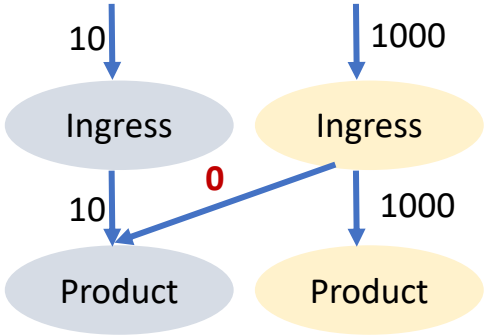
$/ms=1



$/ms=0.0001



$/ms=0

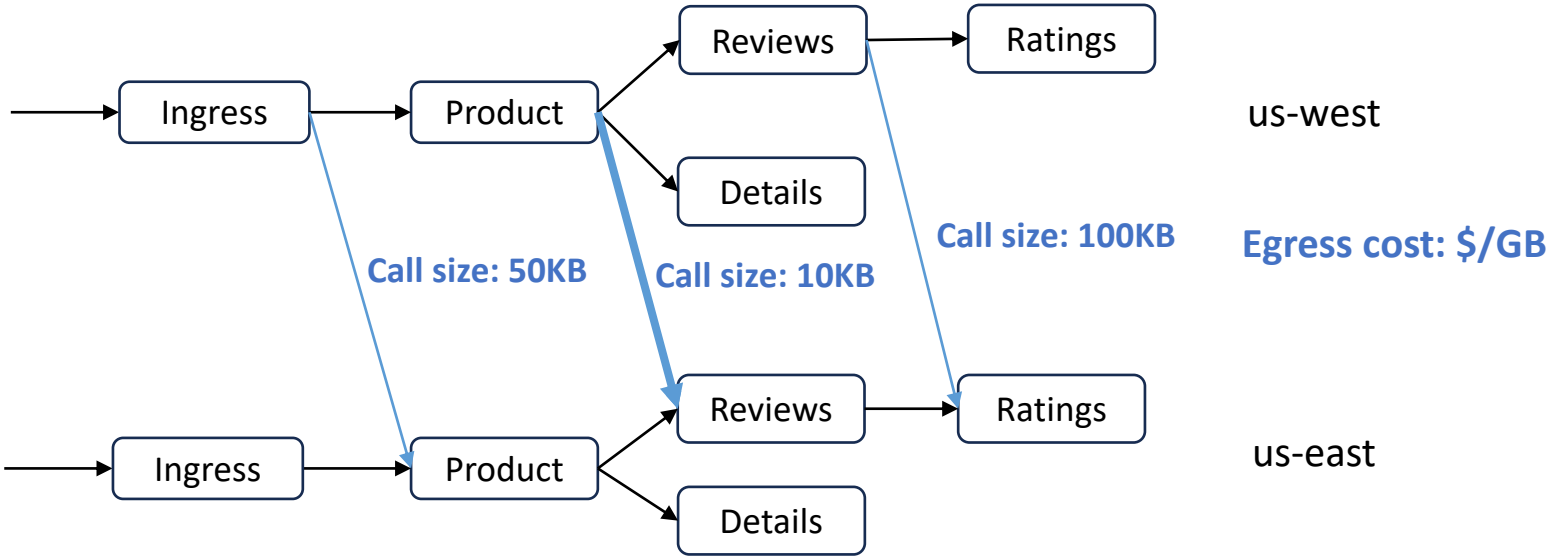


Use case 2: Minimize egress cost

How to make optimal cut?

Service Ratings is not available in cluster 1.

- Service degradation/failure
- GDPR
- security policy, etc.



Use case 2: Minimize egress cost

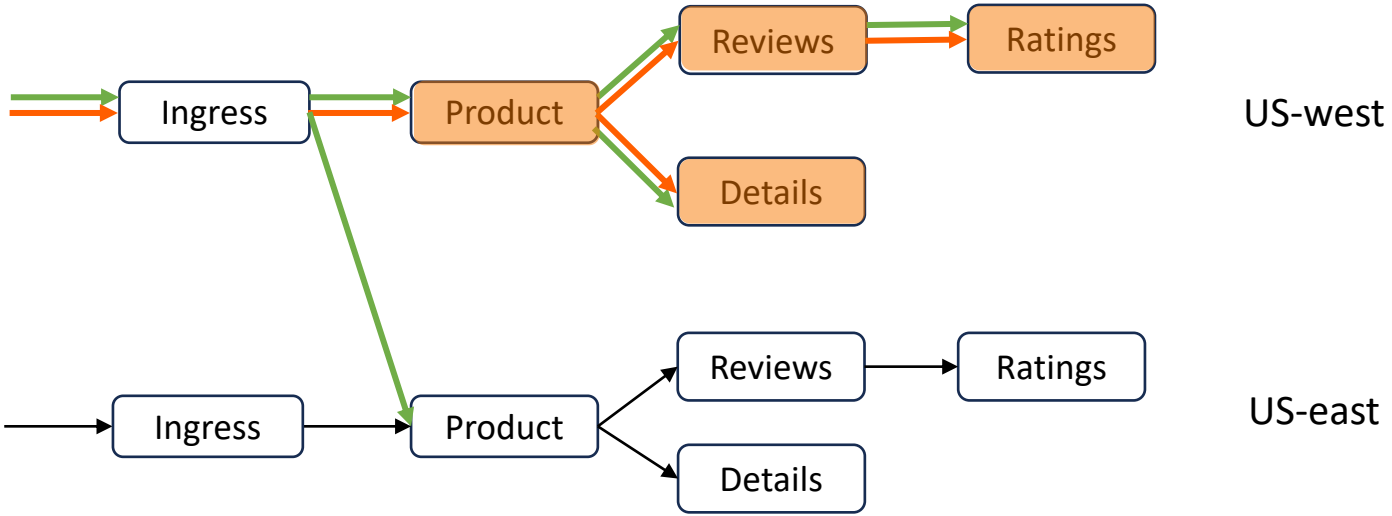
Experiment with Analytics Application

Policy	Bandwidth Cost	Mean Latency (s)
MCLB	$1.03	1.904
Locality Failover	$2.21	4.9185
SLATE	$0.19	0.5828

Use case 3: Per-request decision

Request type 1
- **Compute heavy**
- **Network light**

Request type 2
- **Compute light**
- **Network heavy**



Status and Future plan

- Challenges
  - Latency modeling
  - Call graph prediction at per-request level
- Make the system more compatible
  - Supporting more than two clusters
  - Running more diverse applications
- Deliverable
  - Open source
  - Making it more reliable and pluggable

Demo for use case 1

microburst and latency optimization (4min)