CSCI2952-F

Microservices.. Day 2: Background Continued
Outline

• Containers Versus VMs

• Service Mesh Design Patterns

• API Gateway
  • Motivation
  • Architecture

• eBPF

• YAML
FEATURE

How containers cut server costs at the Financial Times by 80 percent

By Charlotte Jee

Deployment Differences

Containers == process + name space isolation
- Containers are more lightweight than VMs
- Shared OS
- Limited isolation between containers

https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

https://www.researchgate.net/publication/309961613_Containers_and_Virtual_Machines_at_Scale_A_Comparative_Study
Outline

• Containers Versus VMs

• Service Mesh Design Patterns

• API Gateway
  • Motivation
  • Architecture

• eBPF

• YAML
Recall....
ServiceMesh(SM) Dataplane Design Patterns
ServiceMesh (SM) Dataplane Design Patterns

What are the pros/cons of the different approaches?
https://forms.gle/aERbanJoxayLFa176

- SM is in RPC framework
- SM is in a separate user space Proxy
- SM is in a separate user space Proxy
- SM is in the kernel (eBPF)

ServiceMesh (SM) Dataplane Design Patterns

- SM is in RPC framework
  - Used @ FB/Google
  - Tight coupling w/ code
  - Must recompile for changes
  - Low overhead
  - Minimal performance

- SM is in a separate user space Proxy
  - Used @ lyft, DataDog, Tinder...
  - Chic-fila
  - Decouples app code from SM
  - Independent evolution
  - Fault tolerance properties
  - High overheads
  - Performance issues
  - One proxy per container

- SM is in a separate user space Proxy
  - But proxy is shared by all containers
  - Lower overheads
  - Performance issues
  - But lower fault tolerance

- SM is in the kernel (eBPF)
  - Lower overheads
  - Minimal performance issues
  - But lower fault tolerance
  - Limited functionality
  - Requires rewriting everything in eBPF

ServiceMesh (SM) Controlplane Design Patterns

https://logz.io/blog/istio-linkerd-consul-comparison-service-meshes/
- Implements all features
- Hard to configure
- Issues at scale

- Limited feature set
- But easy to operate

- Same binary everywhere
- Easier to deploy
- Well tested functionality


<table>
<thead>
<tr>
<th>Feature</th>
<th>Istio</th>
<th>Linkerd</th>
<th>Consul Connect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics Collection</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Built-In Dashboard</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>TLS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>External Service Support</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Rate Limiting</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Tracing</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Performance Comparison of Popular Implementation

Performance Comparison of Popular Implementation

Which would you use?

- https://forms.gle/6bzK6jJhEsPkiSad9
Which would you use?

https://dzone.com/articles/service-mesh-comparison-istio-vs-linkerd
Roundtable conversation at Kubecon;

Me: which mesh do you use?

AnonSRE: We are thinking of switching to Istio. It has a lot of features?

Me: do you know how you will use them?

AnonSRE: No, but we want to have the option to use them in the future.
Microsoft Debuts Open Service Mesh into a Crowded, Contentious Ecosystem

6 Aug 2020 2:26pm, by Mary Branscombe

Adam Jacob
@adamhjk
Now, while the @Linkerd team should be thrilled that a competitor copied their best in class experience, I suspect they didn’t give attribution? Probably should fix that, @gabi.tv?

oliver @oliv0r
Replying to @oliv0r
LMAO 🥴

12:37 PM · Aug 6, 2020
33 Retweets 136 Likes
See Adam Jacob's other Tweets
Recall....

- How do users connect to your services? What are all the different ways?
- How does this become complicated with a microservice infrastructure?
- What functionality do you need at the edge to interact with these external users?
• API-Gateway unifies access from external API-based communications to internal services
• API-GW provides uniform authentication, verification, auditing and routing
  • One codebase w/ identical policies and enforcement
API Gateway

- API-Gateway unifies access from external API-based communications to internal services
- API-GW provides uniform authentication, verification, auditing and routing
  - One codebase w/ identical policies and enforcement

https://dzone.com/articles/thrift-api-gateway-part-1-bloody-core
Outline

• Service Mesh Design Patterns

• API Gateway
  • Motivation
  • Architecture

• eBPF

• YAML