



DevOps-Continuous Development

Microservice communication



1

Review

-
- This lecture assumes you know
 - DNS
 - Load balancing
 - Failure recovery
 - Canary and A/B testing
 - Monitoring and metrics
 - Pods

2

Overview

- **Discovery**
- Control
- Messaging

Discovery

- Discovery is the process of finding an instance of a desired service.
- For VMs, this is a combination of DNS + Load balancing.
- DNS is a centralized solution to discovery.
- Load balancers (as we described them) balance VMs.

Contextual approach to discovery

- In a contextual approach, there is still a look up by name in a table and the return of an instance location (as in DNS) but
 - There are multiple tables and they are microservices packaged in containers.
 - The tables are specific to a context
- An instance will register (or be registered) in a context dependent look up table.

Context dependencies

- Context dependencies allow for local discovery tables
- “Find an instance of service X within my cluster” takes one request rather than many.
- Canary and A/B testing can be managed by having distinct contexts.
- Requesting service must have its context furnished to it. Through configuration parameters.

Overview

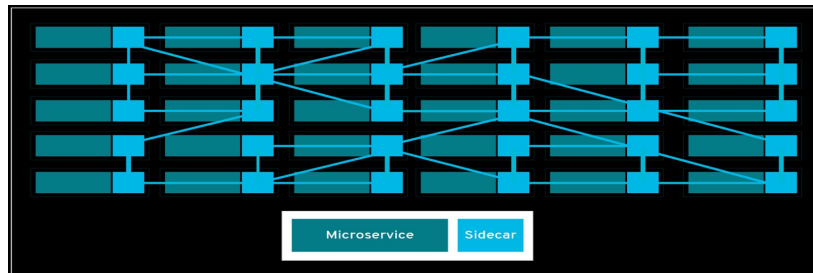
- Discovery
- **Control**
- Messaging

How does one service send a message to another?

- The sending service must discover the location of the recipient service (discovery)
- The message must be routed to the recipient
- Control of routing is done by a service mesh.

Service mesh

- Management functions for microservices (possibly including discovery) are packaged into a service mesh



© Len Bass 2021

9

9

Service Mesh packaging

- A service mesh is a collection of container instances
- Each instance is
 - Attached to an individual microservice instance as a sidecar in a pod
 - The service mesh is the manager for messages to and from that microservice instance

© Len Bass 2021

10

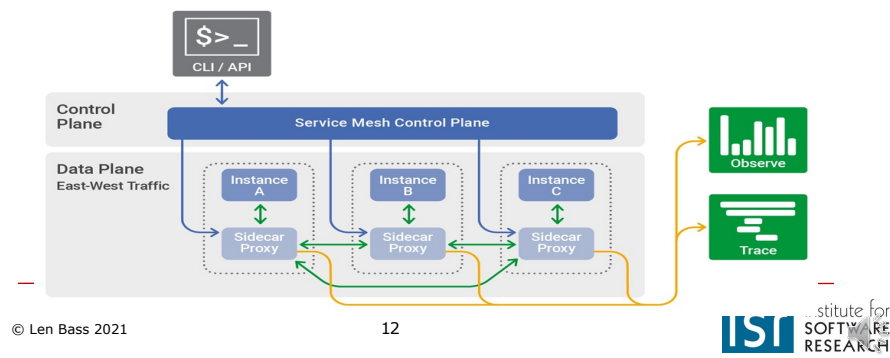
10

Management functions included in a service mesh

- Discovery
- Load balancing,
- Failure recovery,
- Monitoring and metrics.
- Security

Control/data planes

- Management functions are referred to as being in a control plane. Data transfer functions in a data plane.



Overview

- Discovery
- Control
- **Messaging**

Communication between two microservices

- RPC (Remote Procedure Call).
- REST (REpresentational State Transfer)

RPC

- RPC is a remote analog to traditional procedure calls.
 - One procedure calls another with a set of typed arguments, control is transferred to called procedure, and called procedure may respond by returning control
 - Synchronous although multiple threads allow for multiple simultaneous RPCs

Is RPC always remote?

- RPC is designed to support communication across a network.
- If the sender and the recipient are on the same machine, no network needs to be involved – the message is routed directly to the recipient.

gRPC

- gRPC is a binary version of RPC
- Used in conjunction with protocol buffers (discussed shortly)
- Builds on HTTP 2.0
- Typically asynchronous

REST

- Can be used between arbitrary services on the internet
- Designed for web services

REST Characteristics

- The information exchanged is textual – services and methods are accessed by name.
- REST restricts methods to PUT, GET, POST, and DELETE.

Structuring Data

- XML
- JSON
- Protocol Buffers

XML (Extensible Markup Language)

- XML = textual document + schema
- The schema provides tags to describe how to interpret the document
- Tags are used to enclose fields just as in HTML.

JSON (JavaScript Object Notation)

- Textual
- Each data item is designated by a name and a value. E.g.
"addressCountry": "United States",

Protocol Buffers - 1

- Schema defines data types
- Binary format
- A protocol buffer specification is used to specify an interface. Kept in a .proto file
- Language specific compilers used for each side of an interface
- Allows different languages to communicate across a message based interface
- Collection of .proto files defines all of the interfaces and hence all of the microservices.

Protocol Buffers – 2

- Service A written in Java calls Service B written in C
- Interface specification written as .proto file
- Java protocol buffer compiler produces Java procedure interface for Service A
- C protocol buffer compiler produces procedure interface for Service B
- Service A code calls Java procedure interface which sends binary data received by Service B procedure (written in C)

Summary

- Discovery is centralized in microservice applications
- Service mesh manages communication. Individual microservice instances have a service mesh instance attached
- REST and RPC are basic communication protocols.
- XML, JSON and protocol buffers are different structurings of messages.