CNCF SIG Network

Intro & Deep-Dive

Lee Calcote, Layer5

CNCF SIG Network Chair,
CNCF Service Mesh WG Chair

CNCF SIG Network

Introduction

Mission Statement

With an ever steady eye to the needs of workloads and developers who create them and operators who run them, SIG Network's mission is to enable widespread and successful development, deployment and operation of resilient and intelligent network systems in cloud native environments.

In this endeavor, we seek to:

1. Clarify and inform.
2. Collaborate and interrelate.
3. Assist and attract projects.
4. Afford impartial stewardship.

CNCF SIG Network Chairs

KubeCon CloudNativeCon North America 2020 Virtual

Matt Klein

Lyft

@mattklein123

Lee Calcote

Layer5

@lcalcote

Ken Owens

MasterCard

@kenowens12

CNCF SIG Network Projects

KubeCon CloudNativeCon North America 2020 Virtual

As of KubeCon NA 2019

- CNI
- CoreDNS
- Envoy
- gRPC
- Linkerd
- NATS
- Network Service Mesh

As of KubeCon EU 2020

- BFE
- CNI-Genie
- Contour
- Kuma
- Service Mesh Interface

As of KubeCon NA 2020

- Chaos Mesh
- Open Service Mesh

On the horizon:

- Ambassador
- Meshery
- Service Mesh Performance

Working Groups and Papers

Working Groups

- Universal Data Plane API
- Service Mesh Working Group

Whitepapers

- Cloud Native Networking Principles proposed for incorporation into SIG Network.
- Patterns and Reference Architecture

Presentations:

- Moving beyond HTTP: Surveying the State of L7 protocols in the Cloud Native ecosystem

Initiatives

Deep-Dive

Common Goals Across Initiatives

- CNCF labs for at-scale testing
  - Ongoing publication of results

- Service Mesh Patterns
  - A curated collection of best practices and common methodologies.

Service Mesh Patterns

KubeCon CloudNativeCon North America 2020 Virtual

Authors: Lee Calcote, Nic Jackson

[Link to sheet]

Area I: It's a Mesh Out There	Category	Topic
A world of multiple service meshes	Foundational	Why are there service mesh abstractions?
Pattern: How a service mesh empowers an Operator: Retry Budgets	Foundational	
Pattern: How a service mesh empowers a Service Owner	Foundational	
Pattern: How a service mesh empowers a Developer	Foundational	
Pattern: Employing planes of a service mesh	Foundational	
Area II: Patterns of Initialization and Deployment		
Pattern: How to get started with any service mesh; Local Deployment	Deployment	Local Deployment
Pattern: Sidecar Proxies	Deployment	
Pattern: Node Agents	Deployment	
Pattern: Proxyless Service Mesh	Deployment	xDS enabled control Client Libraries with xDS Where does proxyless service mesh fit in, if at all?
Pattern: Passive and Active Health Checking	Deployment	Service Discovery and Health Checking
Pattern: Workload Onboarding and Service Mesh Adoption	Workloads	Adopting service mesh functionality a la carte; Starting small with only a few services.Application migration; Workload onboarding
Pattern: Expanding the Mesh to Brownfield Environments	Workloads	Virtual Machine and Bare Metal; Non-containerized Deployments
Pattern: Segmenting the Monolith (Strangler)	Workloads	Migration
Area III: Patterns of Configuration		
Pattern: Data plane extensibility	Observability	Extending Envoy with WebAssembly (WASM); LuaScript with Nginx; Difference between ESB; using it for user-level authorization;
Pattern: Transparently Proxying TLS	Traffic Management	SNI
Pattern: Foundational Traffic Routing	Traffic Management	Potential for more patterns
Pattern: Local and Global Rate Limiting	Traffic Management	Preventing Denial of Service

Service Mesh Specifications

It's a multi-mesh world

KubeCon CloudNativeCon North America 2020 Virtual

Service Mesh Interface (SMI)

A standard **interface** for service meshes on Kubernetes.

Service Mesh Performance (SMP)

A standard for describing and capturing service mesh **performance**.

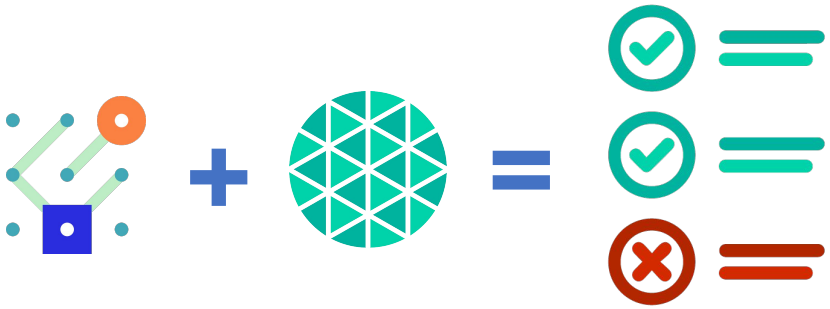
Multi-Vendor Service Mesh Interoperation (Hamlet)

A set of API standards for enabling service mesh **federation**.

Service Mesh Interface Conformance

Validating compliance

KubeCon CloudNativeCon North America 2020 Virtual



- Issue #70
- Design spec
- Overview

Purpose and Scope

- Provide an easy-to-use, service mesh and SMI-specific tool to give service mesh projects and users a suite of repeatable conformance tests.
- All service mesh projects participating in the Service Mesh Interface specification.

Project Goals

- Provide users with a compatibility matrix identifying the SMI features that are supported per service mesh.

Project Objectives

- Define a set of conformance tests and what behavior is expected of a conforming service mesh implementation.
- Built into each participating service mesh project's release tooling.

Validating Conformance

- Conformance to SMI specifications will be done through use of a service mesh's workload.
- A sample application is used as the workload to test: Learn Layer5

Service Mesh Performance

vendor neutral service mesh performance measurement standard

https://smp-spec.io

Directly enables:

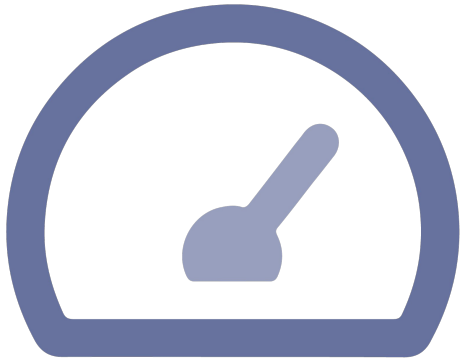
- capturing details of infrastructure capacity, service mesh configuration, and workload metadata.

Facilitates:

- benchmarking of service mesh performance
- exchange of performance information from system-to-system / mesh-to-mesh
- apples-to-apples performance comparisons of service mesh deployments.
- a universal performance index to gauge a service mesh's efficiency against deployments in other organizations' environments.

MeshMark

from the Service Mesh Performance



An open standard for measuring performance of service meshes in context of the value they provide.

Problem:

- Measurement data may not provide a clear and simple picture of how well those applications are performing from a business point of view, a characteristic desired in metrics that are used as key performance indicators.
- Reporting several different kinds of data can cause confusion.

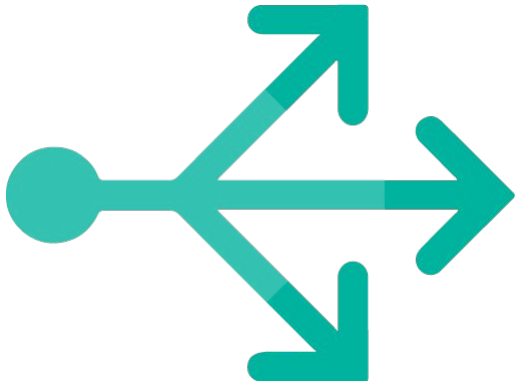
MeshMark:

- Distills a variety of overhead signals and key performance indicators into a simple scale. Reducing measurement data to a single well understood metric is a convenient way to track and report on quality of experience.
- Its purpose is to convert measurements into insights about the value of functions a service mesh is providing.
- It does so by specifying a uniform way to analyze and report on the degree to which measured performance provides user value.

Distributed Performance Analysis

Distributed systems require distributed analysis

KubeCon CloudNativeCon North America 2020 Virtual



layer5.io/projects/distributed-performance-testing

Problem:

- Many performance benchmarks are limited to single instance load generation.This limits the amount of traffic and the variety of behavioral analysis.
- Distributed load testing in parallel poses a challenge when merging results without losing the precision we need to gain insight into the high tail percentiles.

Nighthawk

- a Layer 7 performance characterization tool created by Envoy project.
- a load generator custom-built for data plane proxy testing.

\+

Meshery

- the service mesh management plane
- supports wrk2, fortio, and Nighthawk as single instance load generators.

= Distributed load testing offers insight into system behaviors that arguably more accurately represent real world behaviors of services under load as that load comes from any number of sources.

Call for Participation

- Meet on 1st and 3rd Thursday of every month at 11am Pacific.
- Read: meeting minutes.
- Connect: Slack Channel (#sig-network).
- Join: SIG-Network mailer at lists.cncf.io