



**KubeCon**



**CloudNativeCon**

North America 2018

# gRPC Performance

## Tuning Applications and Libraries

Noah Eisen (email, github, twitter handle: ncteisen)



# Agenda



KubeCon



CloudNativeCon

North America 2018

- gRPC Overview
- Tooling, Benchmarks, and Data
- Tuning the gRPC Library
  - Undoing Death by 1000 Paper Cuts
  - Case Study
- Breaking Down the Layers
- Tuning gRPC Applications
  - Low Hanging Fruit
  - Case Study



KubeCon



CloudNativeCon

North America 2018

# gRPC Overview

# gRPC Overview - History



KubeCon



CloudNativeCon

North America 2018

Borg



Kubernetes



TensorFlow

Blaze



Bazel

Stubby



# gRPC Overview - Basics



KubeCon



CloudNativeCon

North America 2018

gRPC stands for **gRPC Remote Procedure Call**.

★ Unstar

18,115

🔗 Fork

4,189

A **high performance**, open source, standards based, general purpose, polyglot, feature-rich RPC framework.

Actively developed and production-ready.



# gRPC Overview - Generic Stack

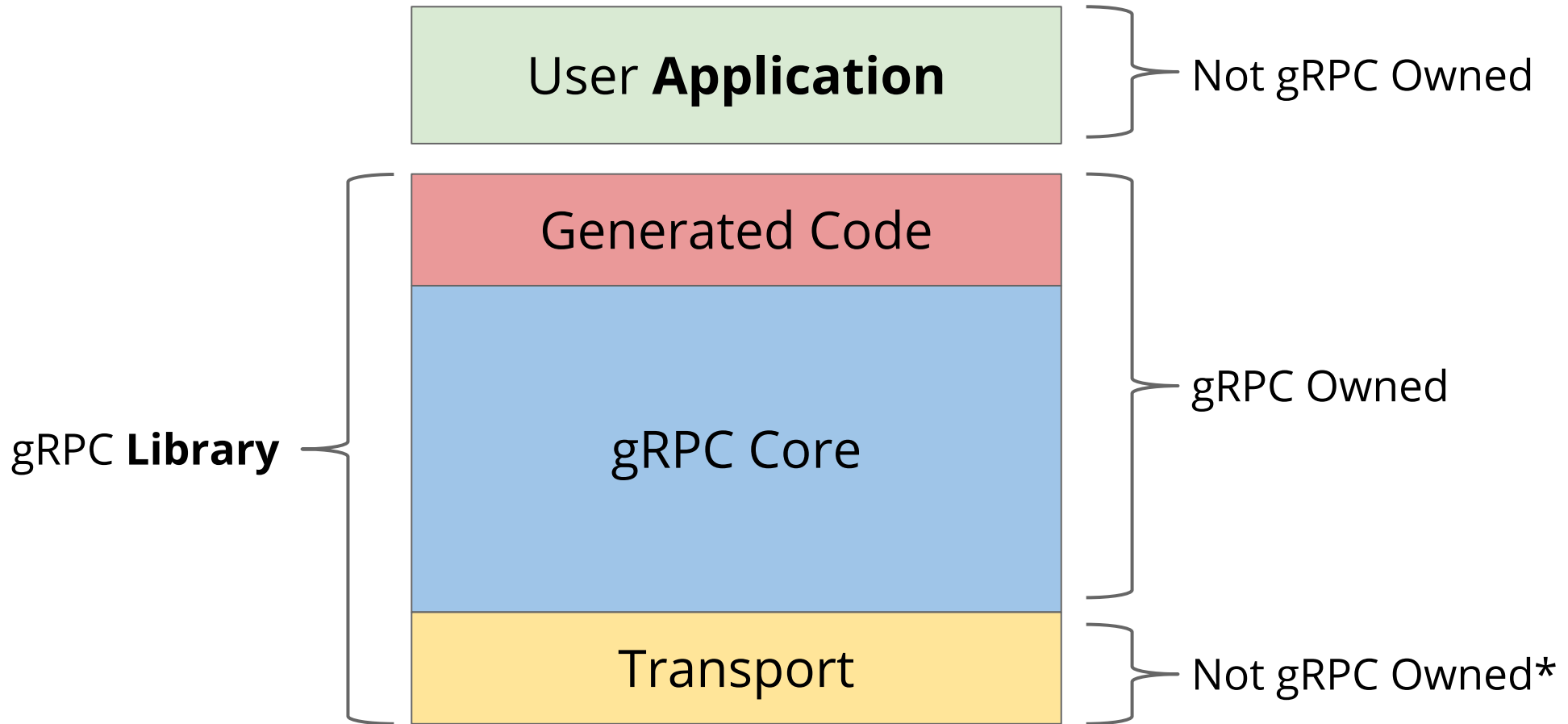


KubeCon



CloudNativeCon

North America 2018



# gRPC Overview - Go Stack

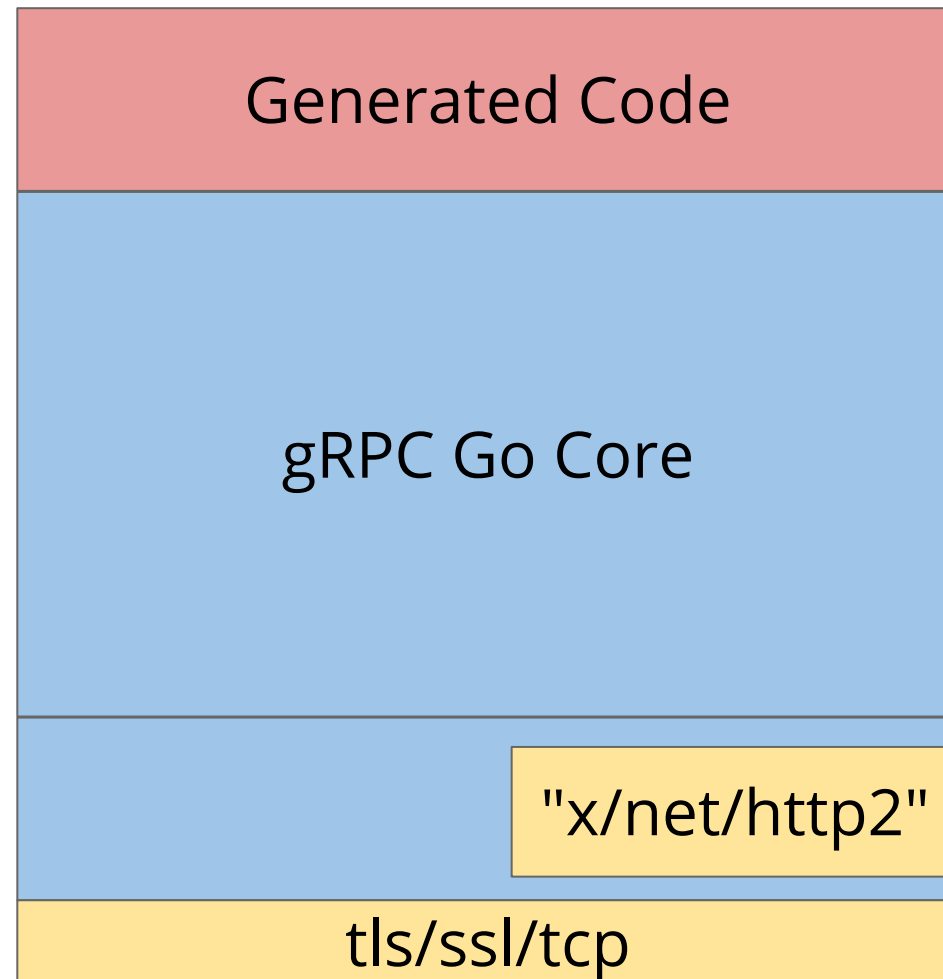


KubeCon



CloudNativeCon

North America 2018



# gRPC Overview - Java Stack

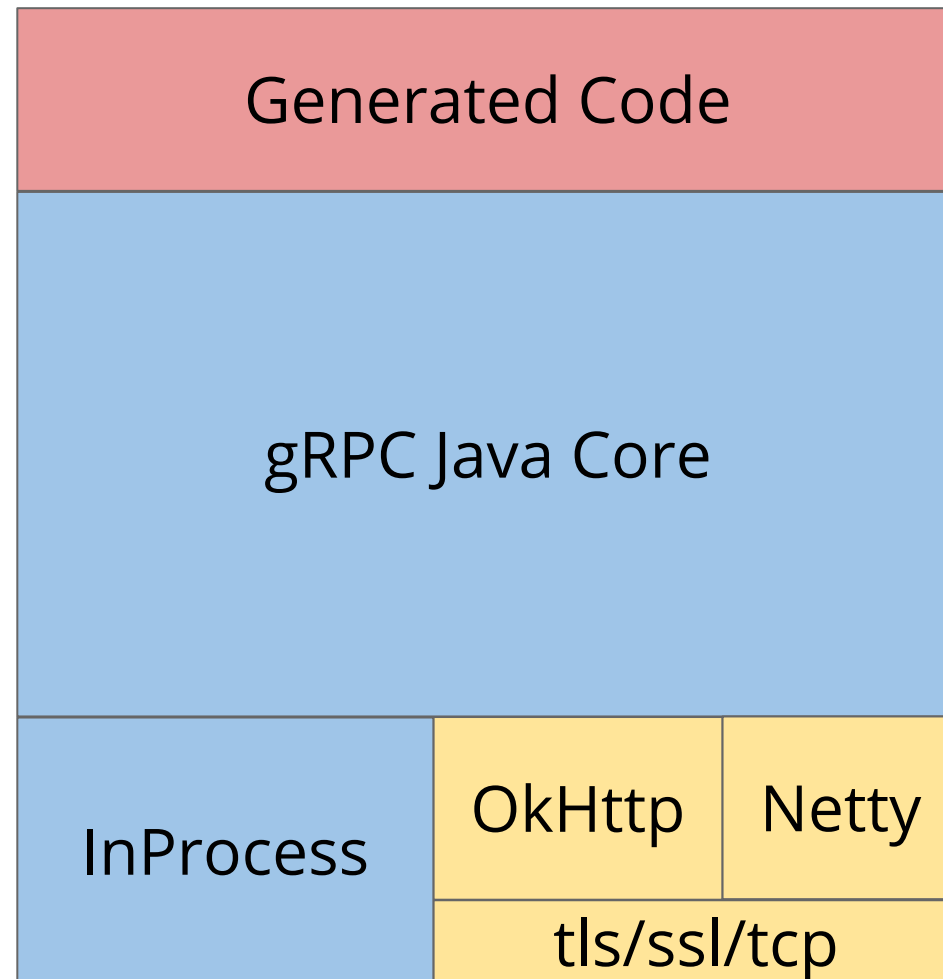


KubeCon



CloudNativeCon

North America 2018





# gRPC Overview - C Stack

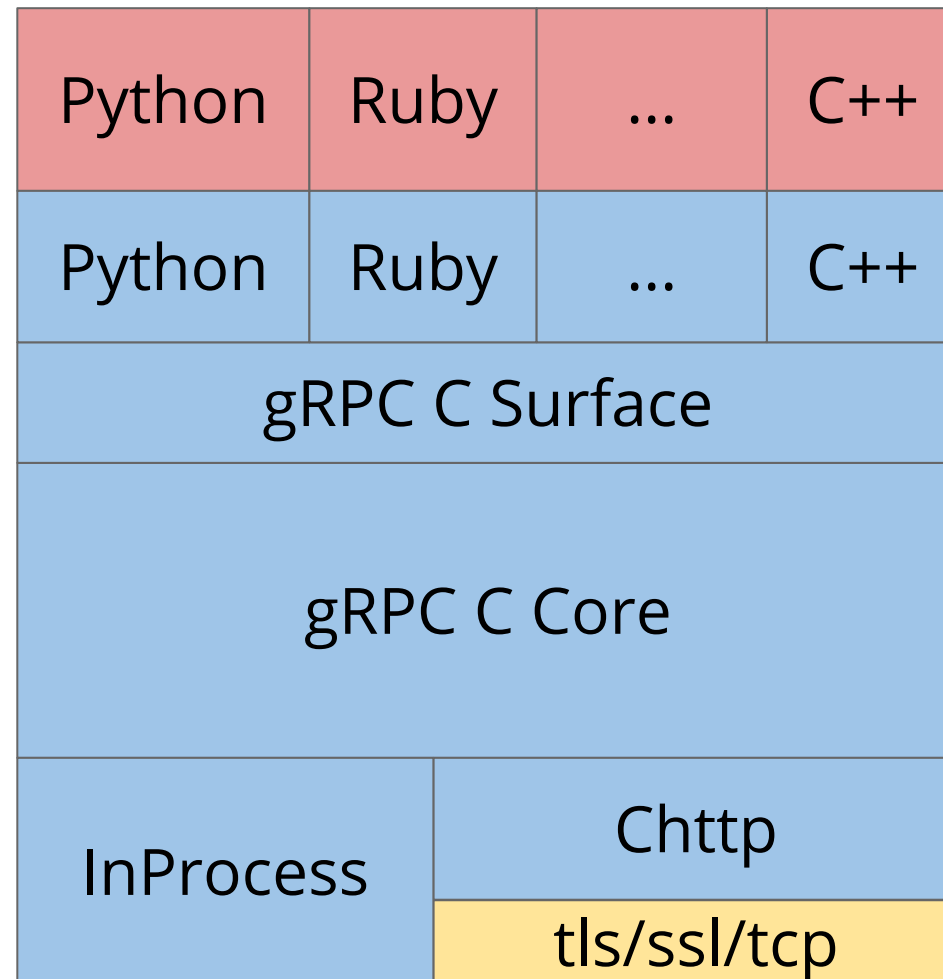


KubeCon



CloudNativeCon

North America 2018



# Tuning Libraries - Key Points



KubeCon



CloudNativeCon

North America 2018

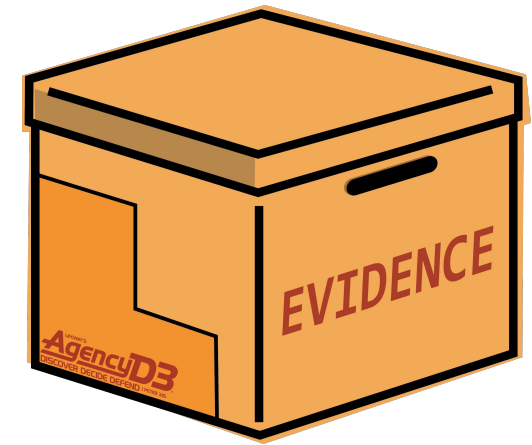
## Tooling



## Benchmarks



## Data



# Tooling



KubeCon

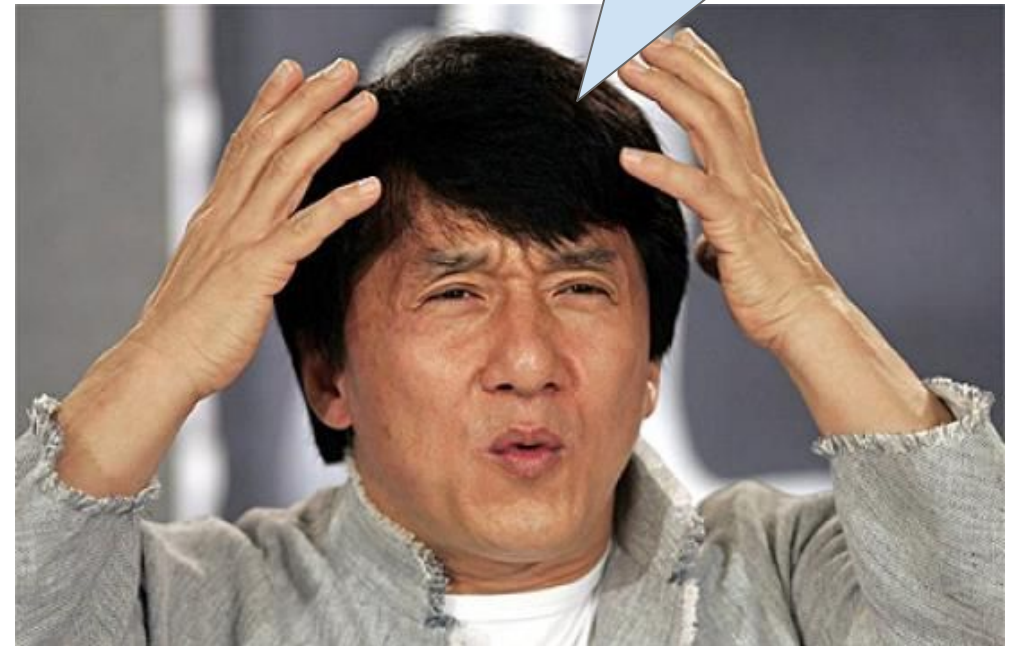


CloudNativeCon

North America 2018

- **In order to optimize, know where to look!**
- Tooling narrows problem scope.
- No such thing as "perfect tool".
- Latency traces.
- CPU profiles.

Where are my  
microseconds  
going??



# Tooling - Other Tools



KubeCon



CloudNativeCon

North America 2018

- Lock contention measuring tools (valgrind, mutrace)
- Custom counters for allocs, atomics.
- Kernel tools:
  - perf (general analysis)
  - strace (syscall introspection)
  - pahole (c++ struct packing)

# Tooling - Bottom Line



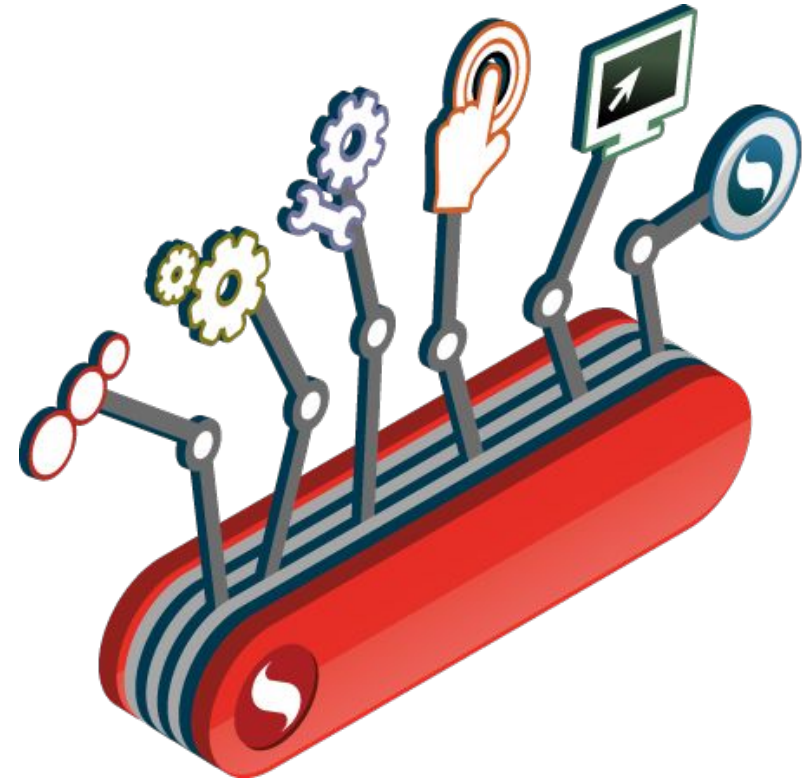
KubeCon



CloudNativeCon

North America 2018

- Obtain an **arsenal of tools**
- Grow your arsenal
- Use tools in conjugation



# Benchmarks



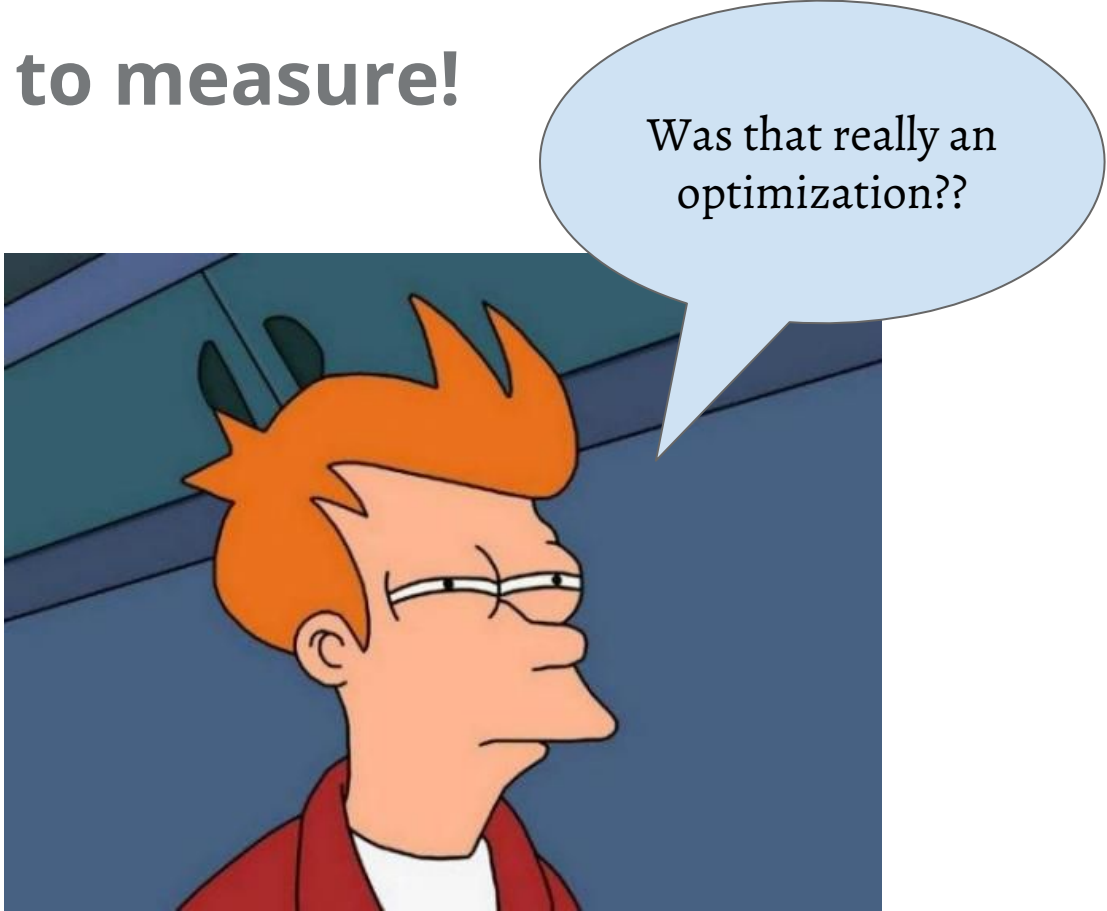
KubeCon



CloudNativeCon

North America 2018

- **In order to optimize, know how to measure!**
- Benchmarks widen scope.



Was that really an optimization??

A cartoon illustration of a man with spiky orange hair and glasses, looking slightly to the right with a questioning expression. A light blue speech bubble is positioned above him, containing the text "Was that really an optimization??". The background is a dark blue gradient with some faint lines.

# Benchmarks - Microbenchmarks



KubeCon



CloudNativeCon

North America 2018

```
static void BM_ErrorCreate(State& state) {  
    while (state.KeepRunning()) {  
        GRPC_ERROR_UNREF(GRPC_ERROR_CREATE("Error"));  
    }  
}  
  
BENCHMARK(BM_ErrorCreate);
```

Run on (12 X 3800 MHz CPU s)

Benchmark	Time	CPU Iterations
BM_ErrorCreate	119 ns	118 ns 5516723

# Benchmarks - Synthetic



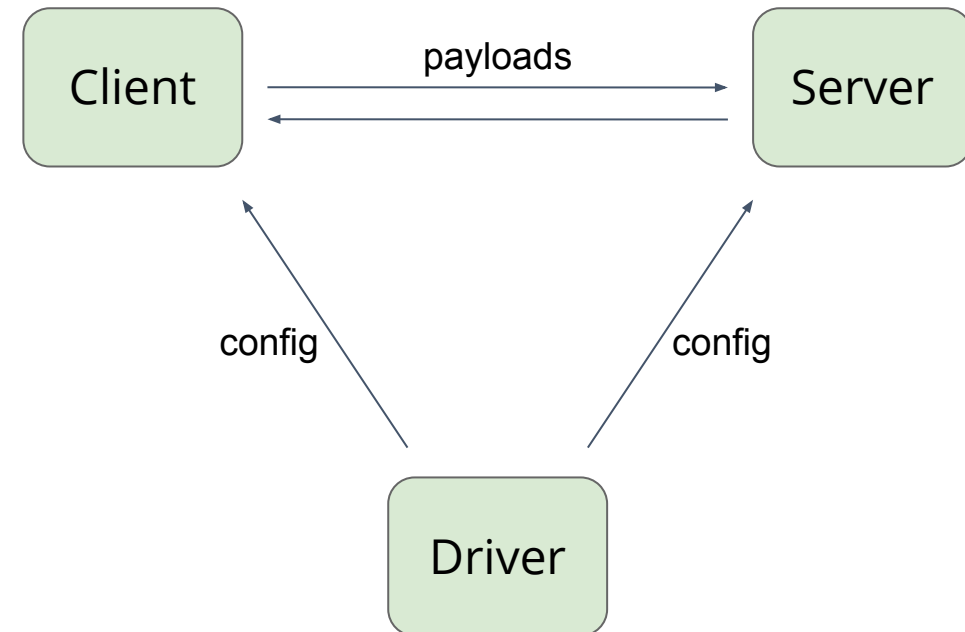
KubeCon



CloudNativeCon

North America 2018

```
{
  "name": "1-channel-1-byte",
  "warmup_seconds": 30,
  "benchmark_seconds": 120,
  "num_servers": 1,
  "server_config": {
    "async_server_threads": 1,
    "server_type": "ASYNC_SERVER"
  },
  "num_clients": 1,
  "client_config": {
    "client_type": "ASYNC_CLIENT",
    "payload_config": {
      "simple_params": {
        "resp_size": 1,
        "req_size": 1
      }
    },
    "client_channels": 1,
    "async_client_threads": 1,
    "rpc_type": "UNARY",
    "load_params": {
      "closed_loop": {}
    }
  }
}
```





# Benchmarks - Application



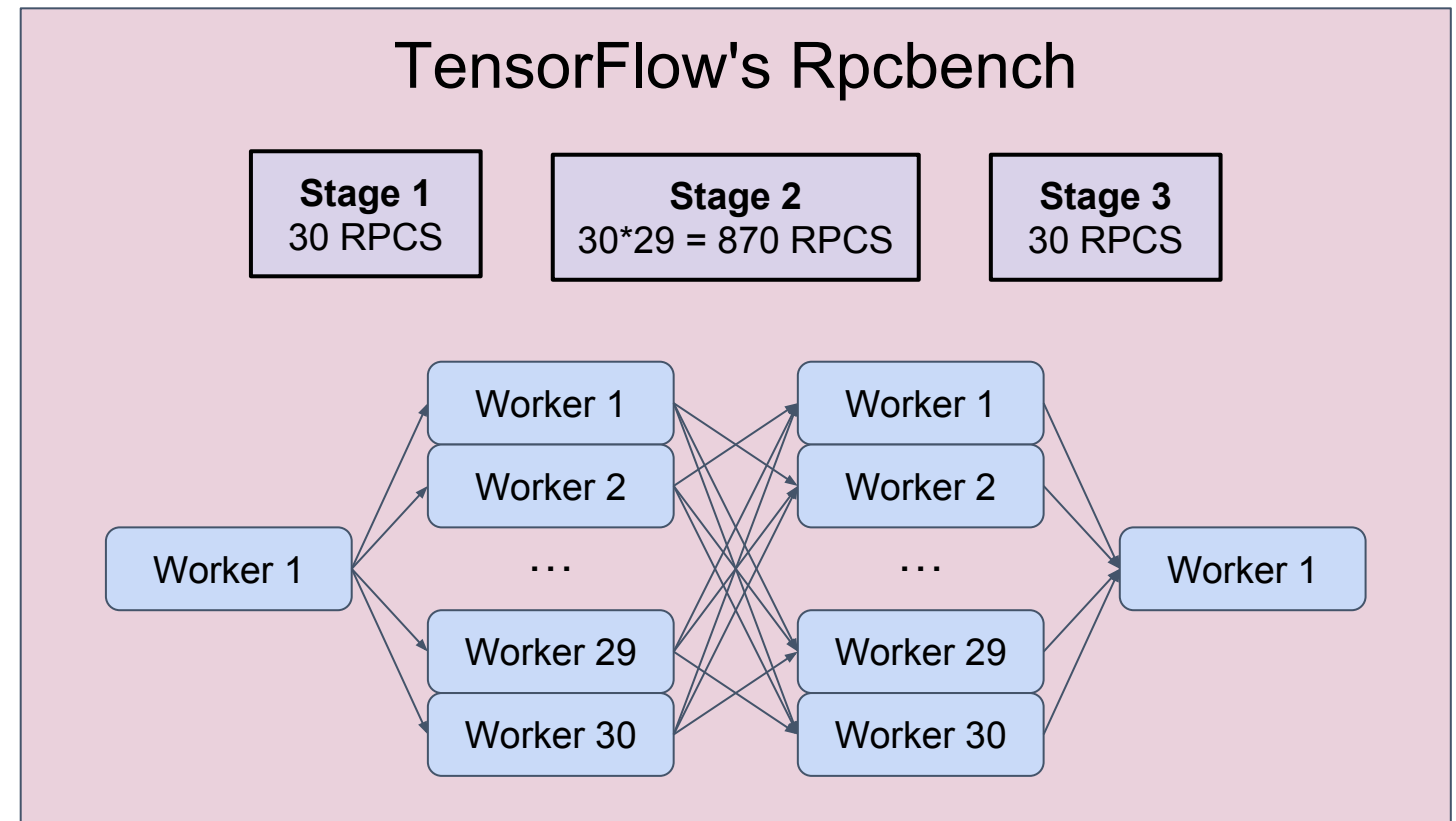
KubeCon



CloudNativeCon

North America 2018

- Written via another team's API.
- Exercises the stack in new ways.
- Only applies to libraries.



# Data



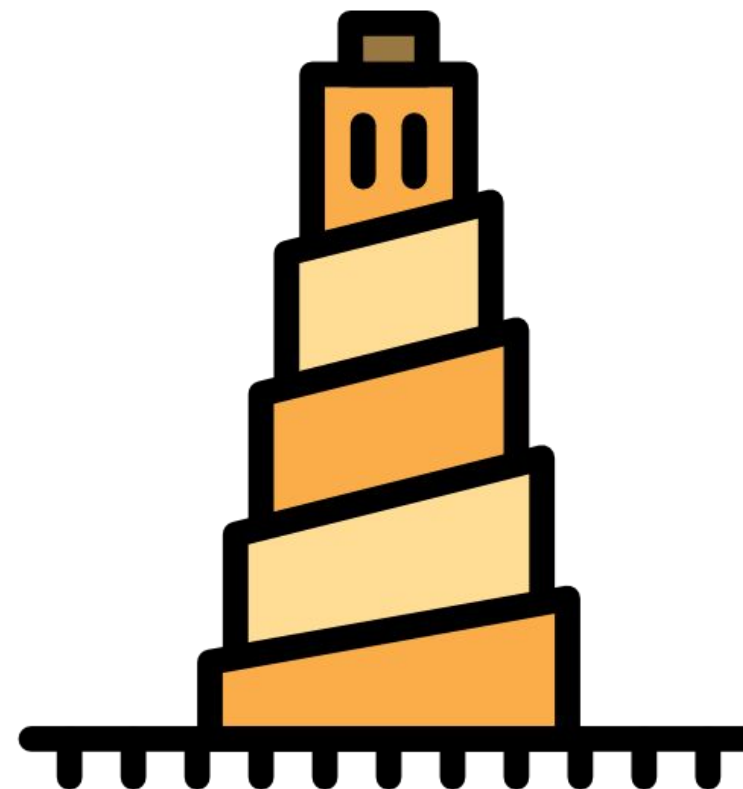
KubeCon



CloudNativeCon

North America 2018

- Team needs a *lingua franca*
- Optimizations come with:
  - data from tooling
  - data from benchmarks



# Narrowing and Widening



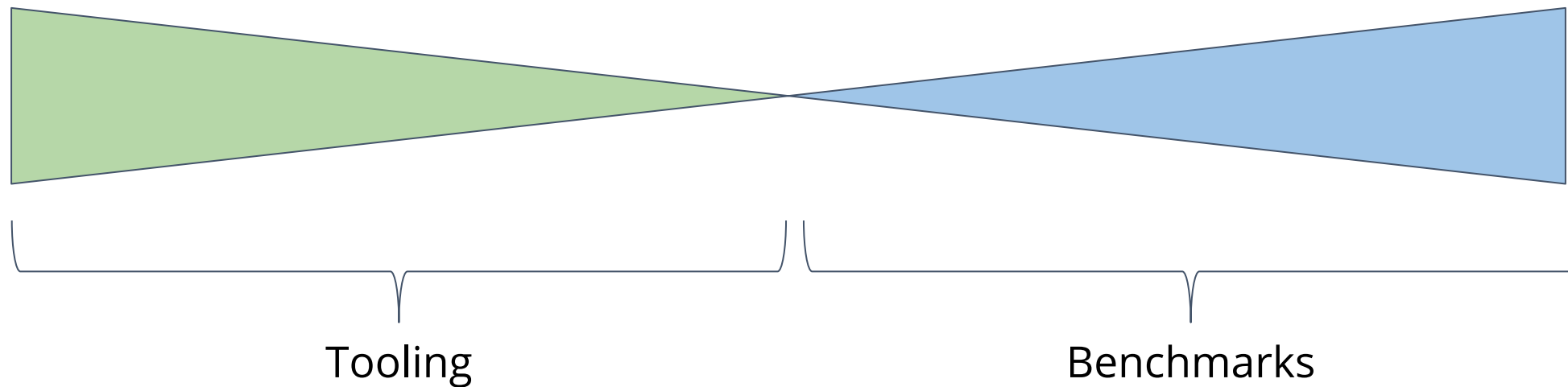
KubeCon



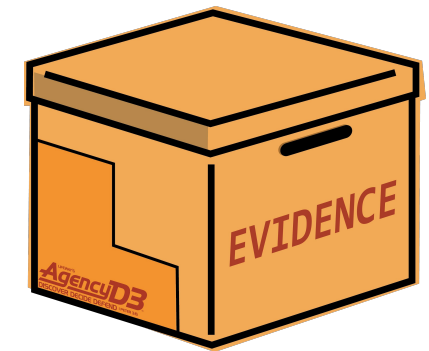
CloudNativeCon

North America 2018

- Tooling narrows scope
- Benchmarks widen scope
- End result is **data**



Data





KubeCon



CloudNativeCon

North America 2018

# Tuning the gRPC Library

# Undoing Death by 1000 Paper Cuts



KubeCon



CloudNativeCon

North America 2018

What to do once the "low hanging fruit" has been taken?

- Features can cause small regressions.
- Sometimes, *below margin of detection*.
- Consistent, slow, degradation of performance.

**How do we reverse this process?**

# Undoing Death by 1000 Paper Cuts



KubeCon



CloudNativeCon

North America 2018

- New benchmark: **Minimal RPC**
  - Ping pong of 1 byte payloads
  - No security
  - No stats or tracing
  - Focused on median latency
- New tooling to use
- Noise reduction

# Breaking Down the Layers



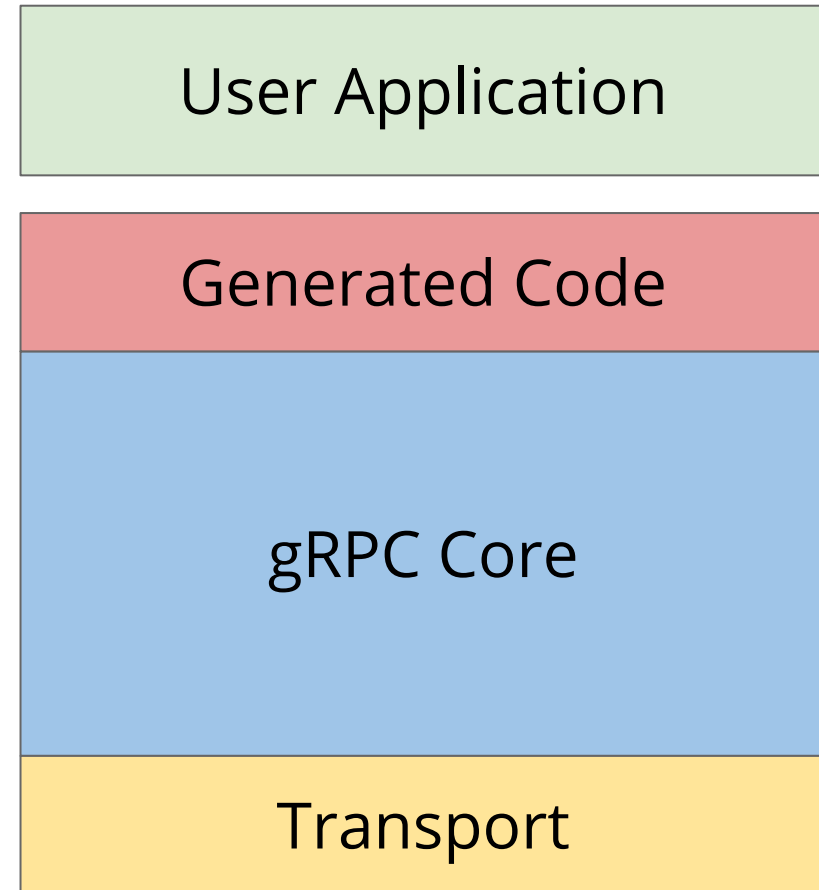
KubeCon



CloudNativeCon

North America 2018

- Tuning below:
  - Contributions to Netty, OkHttp.
  - Contributions to x/net/http.
  - Tuning work with TCP team.
- Tuning above:
  - Next part of this talk.





KubeCon



CloudNativeCon

North America 2018

# Tuning gRPC Applications



# Low Hanging Fruit



KubeCon



CloudNativeCon

North America 2018

- All Language Stacks
  - Reduce allocations
  - Reduce copies
  - Reduce syscalls
  - Reduce contention
- Java Stack
  - Use async API
  - Tune thread pools
  - Tune Netty direct memory
  - Consider Netty Epoll / kqueue
- C++ Stack:
  - Use async API
  - Tune threading model
  - # of completion queues
  - # of outstanding RPCs
- Go Stack
  - Parallelize with goroutines
  - Tune Read/Write buffer size
  - # of outstanding RPCs

# And of Course



KubeCon



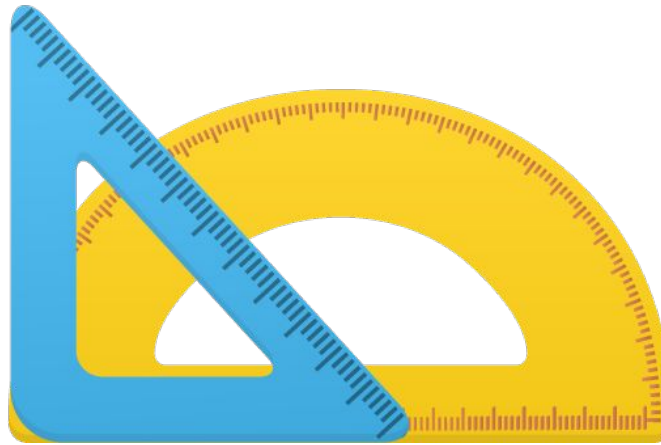
CloudNativeCon

North America 2018

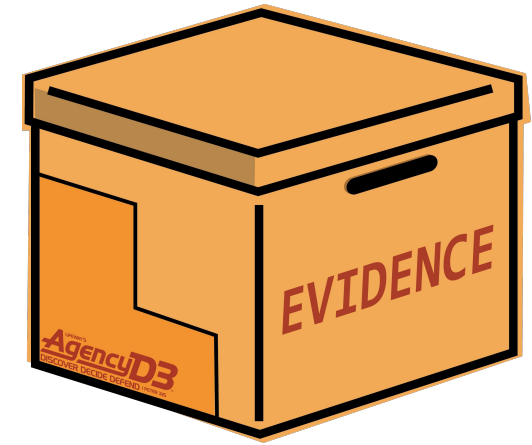
## Tooling



## Benchmarks



## Data



# Case Study



KubeCon



CloudNativeCon

North America 2018

- Distributed TensorFlow
  - Needs network layer
  - Uses gRPC
- **Goal:** improve TensorFlow-over-gRPC performance



gRPC

# Case Study



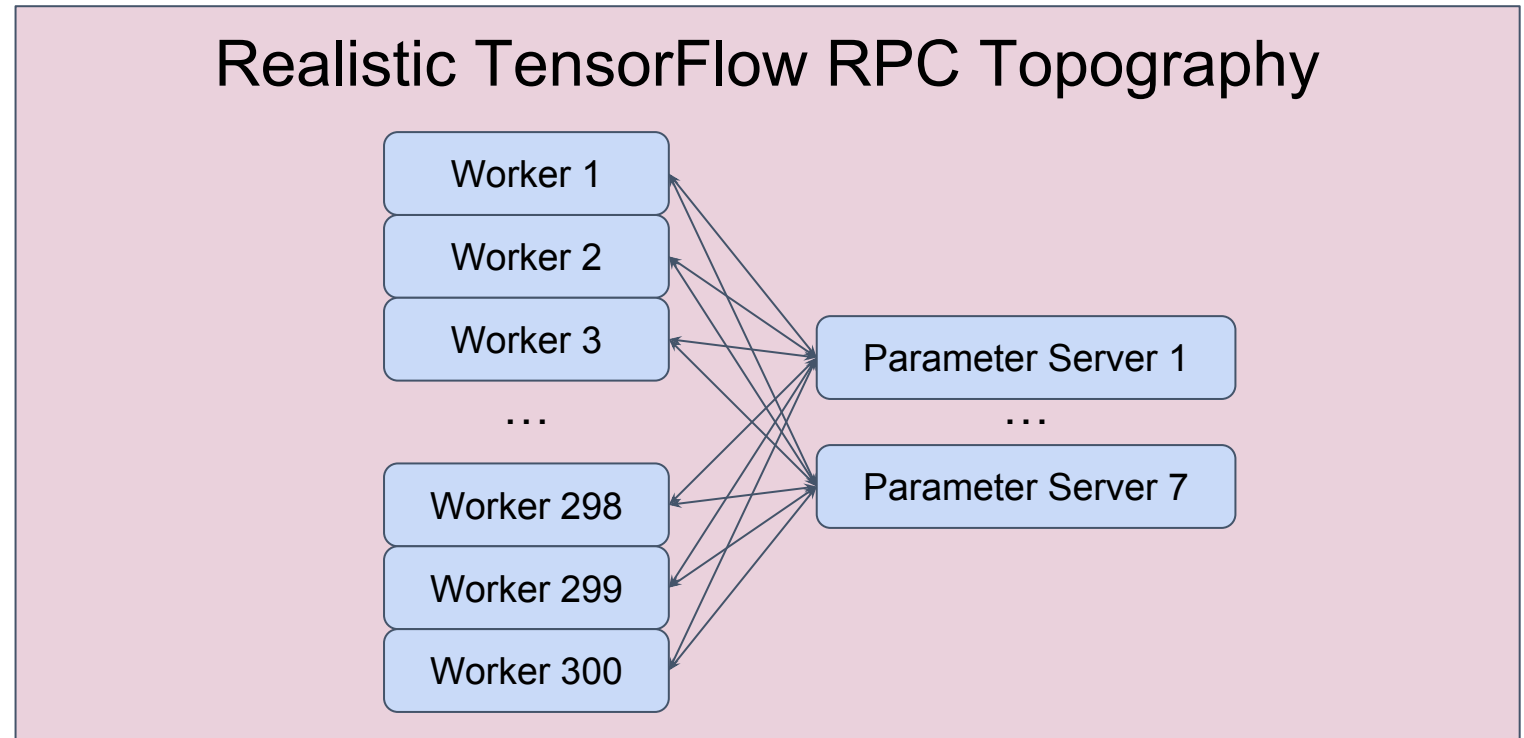
KubeCon



CloudNativeCon

North America 2018

- Benchmarks:
  - Rpcbench
  - Real TensorFlow training tasks
- Impactful changes:
  - better threading
  - serialization threadpool



# Breaking Down the Layers (again)



KubeCon



CloudNativeCon

North America 2018

- Tuning below:
  - Contribute to gRPC!



Your Application

A light green rectangular box representing the application layer.

gRPC Library

A diagram showing three stacked layers: a red top layer, a blue middle layer labeled 'gRPC Library', and a yellow bottom layer.

# Thank You!



KubeCon



CloudNativeCon

North America 2018

- gRPC Resources:
  - <http://grpc.io>
  - <http://grpc.io/contribute>
  - <https://github.com/grpc>
  - <https://github.com/grpc-ecosystem>
- Personal Contact:
  - Email: [ncteisen@google.com](mailto:ncteisen@google.com)
  - GitHub: <https://github.com/ncteisen>
  - Website: <http://noaheisen.com>





**KubeCon**

**CloudNativeCon**

————— **North America 2018** —————





**KubeCon**



**CloudNativeCon**

North America 2018

# Appendix



# Optimization: DoThenAsyncNext



KubeCon



CloudNativeCon

North America 2018

- **Author:** [kpayson64@](#)
- **Change:** [#13084](#)
- **Location:** gRPC Core.
- **Context:** gRPC has a asynchronous completion queue API. Work is placed on the queue by the application, driven by calls to AsyncNext, and then completion events are returned to application.
- **Optimization:** New API in which application can pass a lambda to be executed before AsyncNext. If this lambda triggers a completion event, it is returned by the call to AsyncNext.
- **TL;DR:** Reduced thread hops in a common case.

# Optimization: TF Threading



KubeCon



CloudNativeCon

North America 2018

- **Author:** [ncteisen@](#)
- **Change:** [0d5fb10](#)
- **Location:** TensorFlow application layer.
- **Context:** TensorFlow has GrpcWorker class, which is responsible for encapsulating the gRPC network layer from the TensorFlow application.
- **Optimization:** Allow multiple threads to service the GrpcWorker's completion queues.
- **TL;DR:** More parallelism.
-

# Optimization: Epoll Exclusive



KubeCon



CloudNativeCon

North America 2018

- **Author:** [ctiller@](#)
- **Change:** [#12789](#)
- **Location:** gRPC Core.
- **Context:** gRPC has an internal polling system to efficiently interact with network I/O. It has gone through several iterations and optimizations.
- **Optimization:** New polling system, epoll, that relies on the EPOLLEXCLUSIVE flag for epoll\_ctl.
- **TL;DR:** Thread are woken up more efficiently.

# Optimizations to the Minimal RPC



KubeCon



CloudNativeCon

North America 2018

- hcaseyal@
  - #15839, #15879, #15883 (moving allocations to call arena)
- kpayson64@
  - #13947 (adds fd cache to avoid epoll\_ctl)
- ncteisen@
  - #15578 (compile out spammy tracer)
- yashykt@
  - #15280 (compile out stats machinery in opt builds)
  - #15200 (adds new closure scheduling mechanism)
  - #15044 (adds compiler hints)