Go Cloud

Go 1.11 Release Party in Tokyo Aug 22nd, 2018

Yoshi Yamaguchi (@ymotongpoo)

Google Cloud

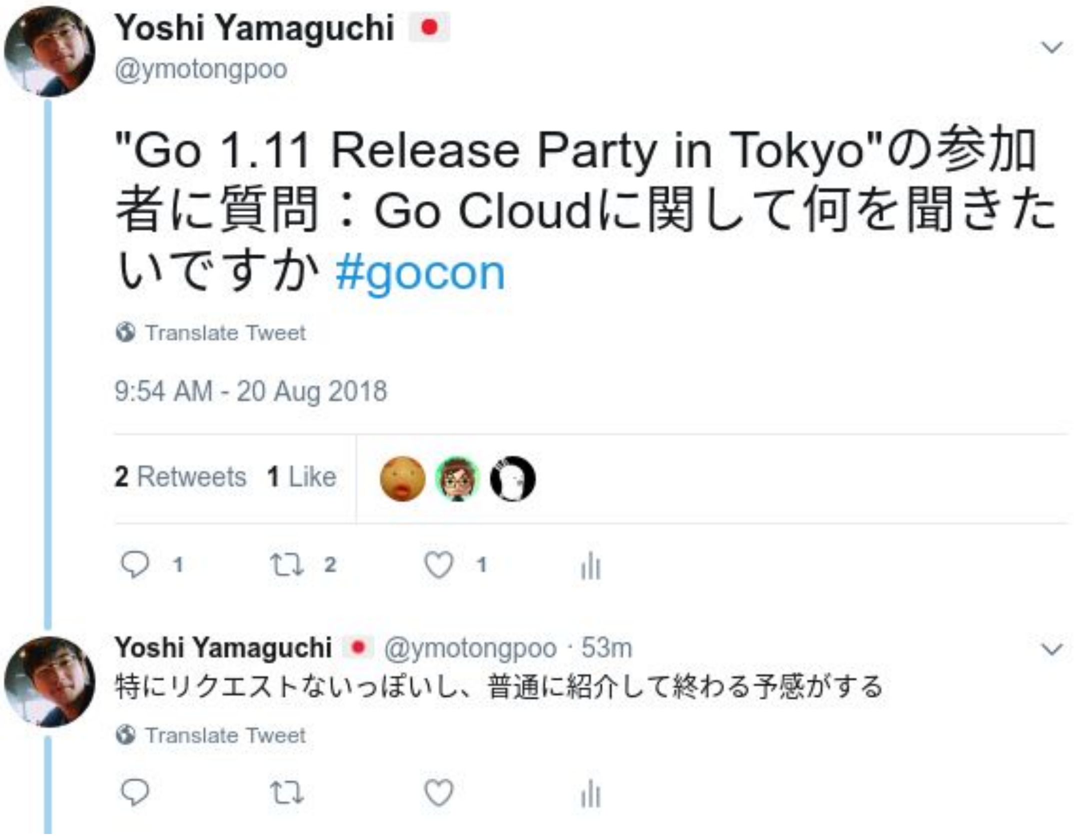
Who are you?

Yoshi Yamaguchi (@ymotongpoo)

Developer Advocate for Google Cloud

- Observability (Stackdriver)
- Go

Google Cloud



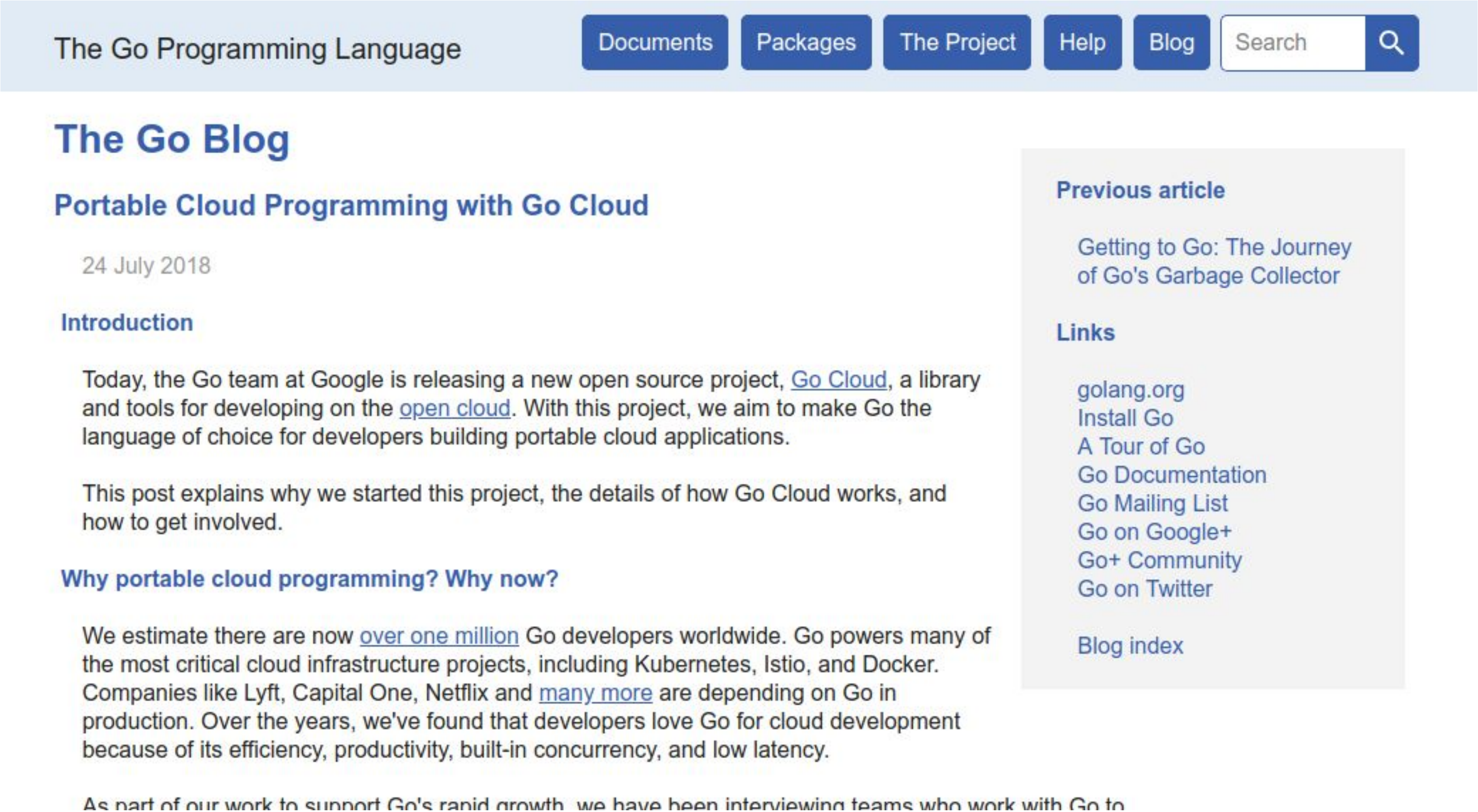
Yoshi Yamaguchi 🇯🇵
@ymotongpoo

"Go 1.11 Release Party in Tokyo"の参加者に質問：Go Cloudに関して何を聞きたいですか #gocon

9:54 AM - 20 Aug 2018

2 Retweets 1 Like

Yoshi Yamaguchi 🇯🇵 @ymotongpoo · 53m
特にリクエストないっぽいし、普通に紹介して終わる予感がする



The Go Blog

Portable Cloud Programming with Go Cloud

24 July 2018

Introduction

Today, the Go team at Google is releasing a new open source project, Go Cloud, a library and tools for developing on the open cloud. With this project, we aim to make Go the language of choice for developers building portable cloud applications.

This post explains why we started this project, the details of how Go Cloud works, and how to get involved.

Why portable cloud programming? Why now?

We estimate there are now over one million Go developers worldwide. Go powers many of the most critical cloud infrastructure projects, including Kubernetes, Istio, and Docker. Companies like Lyft, Capital One, Netflix and many more are depending on Go in production. Over the years, we've found that developers love Go for cloud development because of its efficiency, productivity, built-in concurrency, and low latency.

As part of our work to support Go's rapid growth, we have been interviewing teams who work with Go to

Previous article

Getting to Go: The Journey of Go's Garbage Collector

Links

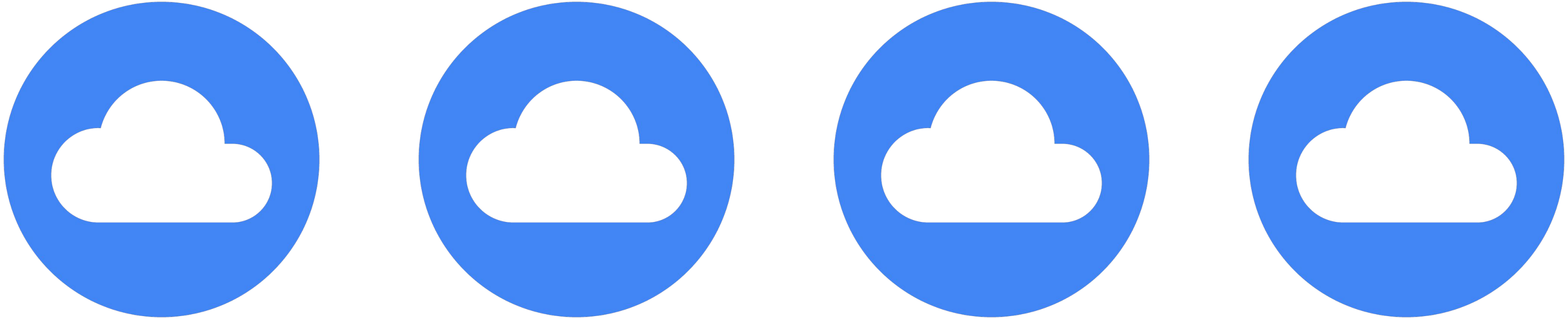
golang.org
Install Go
A Tour of Go
Go Documentation
Go Mailing List
Go on Google+
Go+ Community
Go on Twitter

Blog index

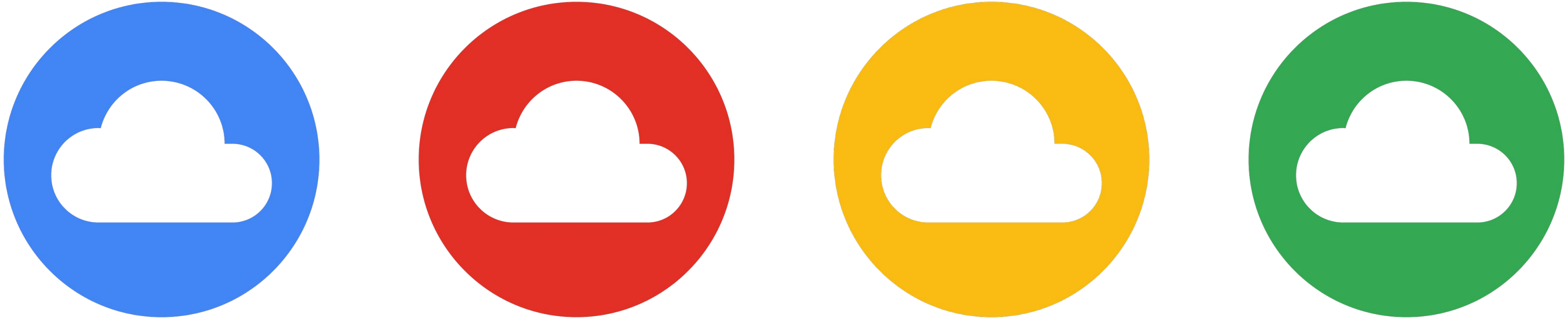
Google Cloud

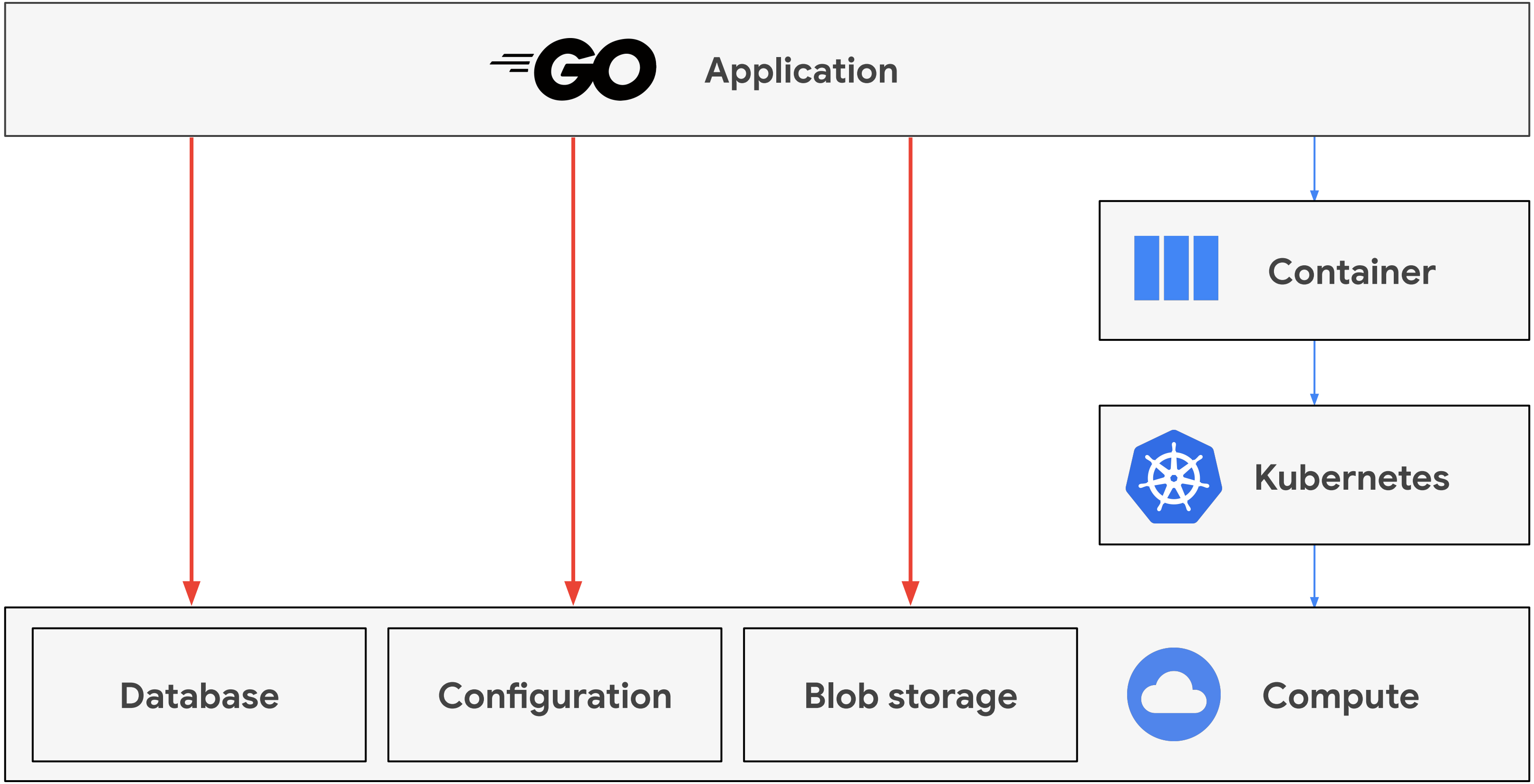
blog.golang.org/go-cloud

Each cloud is **similar** at a high level.



Each cloud is **different** at a low level.




GO
Application
Container
Kubernetes
Database
Configuration
Blob storage
Compute

Demo

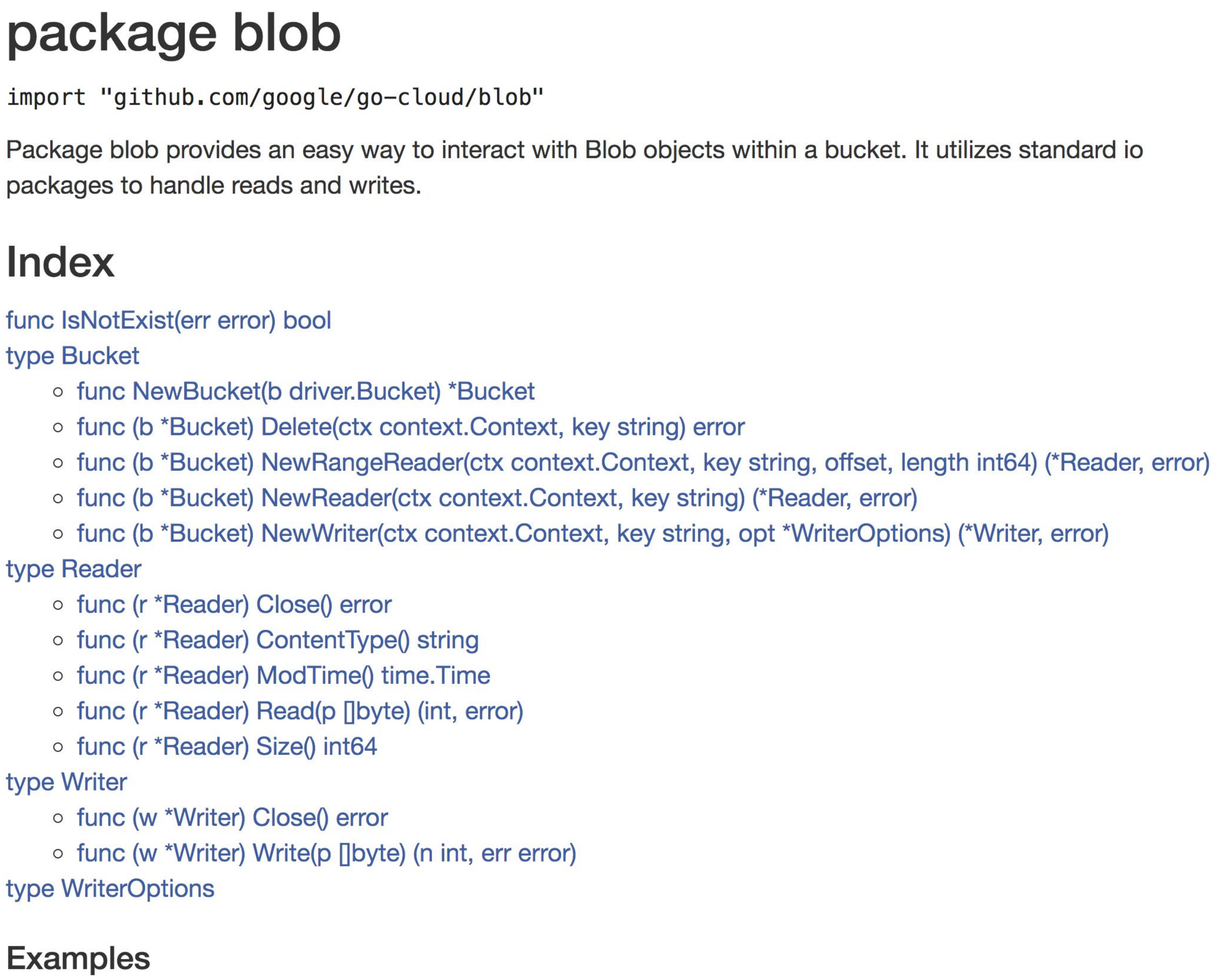
Demo scenario

- Build demo with Go 1.11 using module
- Show source code of demo
- Run build binary

Supported APIs

- Blob storage
- Runtime configuration
- MySQL database access
- Request logging
- Health checking
- Tracing

ver 0.1.1



package blob

```
import "github.com/google/go-cloud/blob"
```

Package blob provides an easy way to interact with Blob objects within a bucket. It utilizes standard io packages to handle reads and writes.

Index

func IsNotExist(err error) bool

type Bucket

- func NewBucket(b driver.Bucket) *Bucket
- func (b *Bucket) Delete(ctx context.Context, key string) error
- func (b *Bucket) NewRangeReader(ctx context.Context, key string, offset, length int64) (*Reader, error)
- func (b *Bucket) NewReader(ctx context.Context, key string) (*Reader, error)
- func (b *Bucket) NewWriter(ctx context.Context, key string, opt *WriterOptions) (*Writer, error)

type Reader

- func (r *Reader) Close() error
- func (r *Reader) ContentType() string
- func (r *Reader) ModTime() time.Time
- func (r *Reader) Read(p []byte) (int, error)
- func (r *Reader) Size() int64

type Writer

- func (w *Writer) Close() error
- func (w *Writer) Write(p []byte) (n int, err error)

type WriterOptions

Examples

Currently it supports **Google Cloud Platform** and **Amazon Web Services**.

Appendix

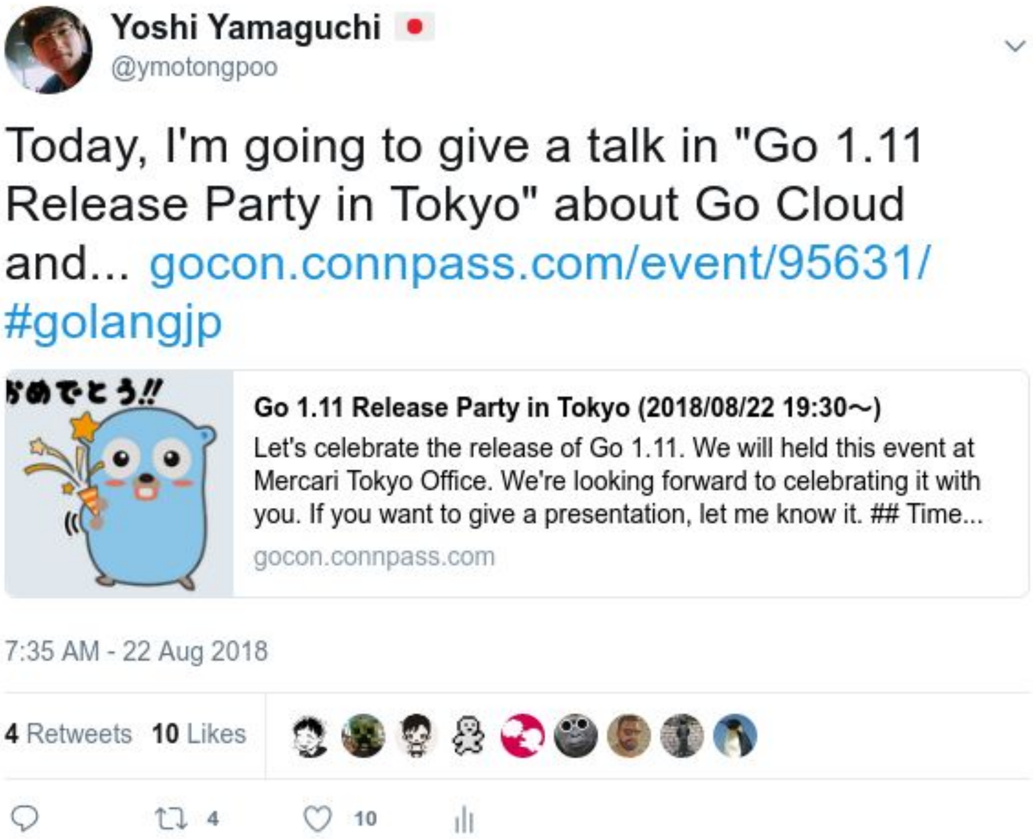
Blog post/Presentation

blog.golang.org/go-cloud

youtu.be/_2ZwhvIkgek

Repository

github.com/google/go-cloud



Yoshi Yamaguchi
@ymotongpoo

Today, I'm going to give a talk in "Go 1.11 Release Party in Tokyo" about Go Cloud and... gocon.connpass.com/event/95631/ #golangjp

Go 1.11 Release Party in Tokyo (2018/08/22 19:30〜)

Let's celebrate the release of Go 1.11. We will held this event at Mercari Tokyo Office. We're looking forward to celebrating it with you. If you want to give a presentation, let me know it. ## Time...

gocon.connpass.com

7:35 AM - 22 Aug 2018

4 Retweets 10 Likes

4 10

and...?

OpenCensus and Stackdriver Tracing

Go 1.11 Release Party in Tokyo   Aug 22nd, 2018

Yoshi Yamaguchi (@ymotongpoo)

Google Cloud

You can't tell where a program is going to spend its time. **Bottlenecks occur in surprising places**, so don't try to second guess and put in a speed hack until you've proven that's where the bottleneck is.

Rob Pike
"Complexity", Notes on Programming in C

Measure. Don't tune for speed until you've measured, and even then don't unless one part of the code overwhelms the rest.

Rob Pike
"Complexity", Notes on Programming in C

Go standard packages for Observability

Tracing

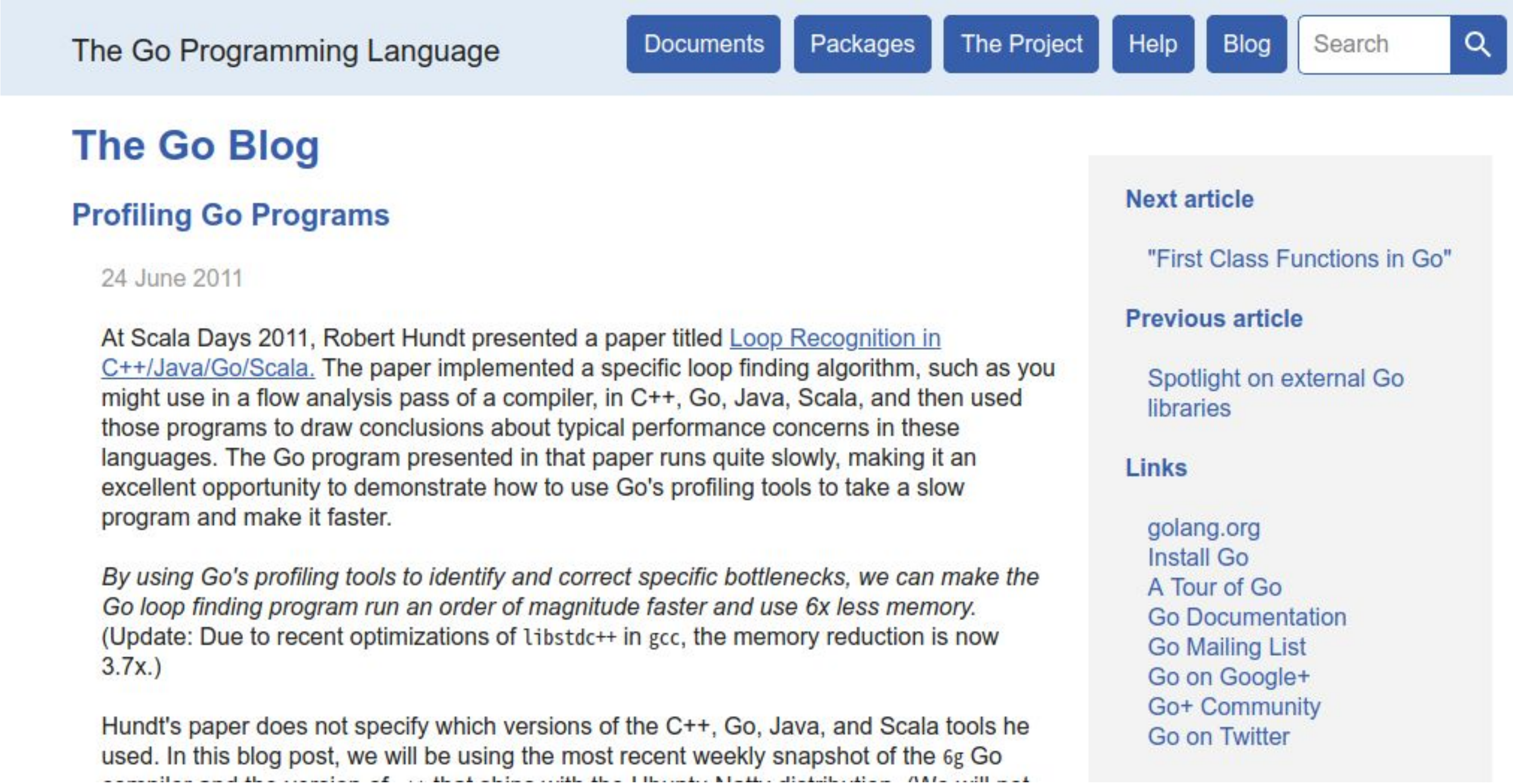
- runtime/trace
- net/http/httptrace

```
go tool trace trace.out
```

Profiling

- runtime/pprof
- net/http/pprof

```
go tool pprof myapp myapp.prof
```



The Go Blog

Profiling Go Programs

24 June 2011

At Scala Days 2011, Robert Hundt presented a paper titled Loop Recognition in C++/Java/Go/Scala. The paper implemented a specific loop finding algorithm, such as you might use in a flow analysis pass of a compiler, in C++, Go, Java, Scala, and then used those programs to draw conclusions about typical performance concerns in these languages. The Go program presented in that paper runs quite slowly, making it an excellent opportunity to demonstrate how to use Go's profiling tools to take a slow program and make it faster.

By using Go's profiling tools to identify and correct specific bottlenecks, we can make the Go loop finding program run an order of magnitude faster and use 6x less memory. (Update: Due to recent optimizations of `libstdc++` in `gcc`, the memory reduction is now 3.7x.)

Hundt's paper does not specify which versions of the C++, Go, Java, and Scala tools he used. In this blog post, we will be using the most recent weekly snapshot of the `6g` Go

Next article

"First Class Functions in Go"

Previous article

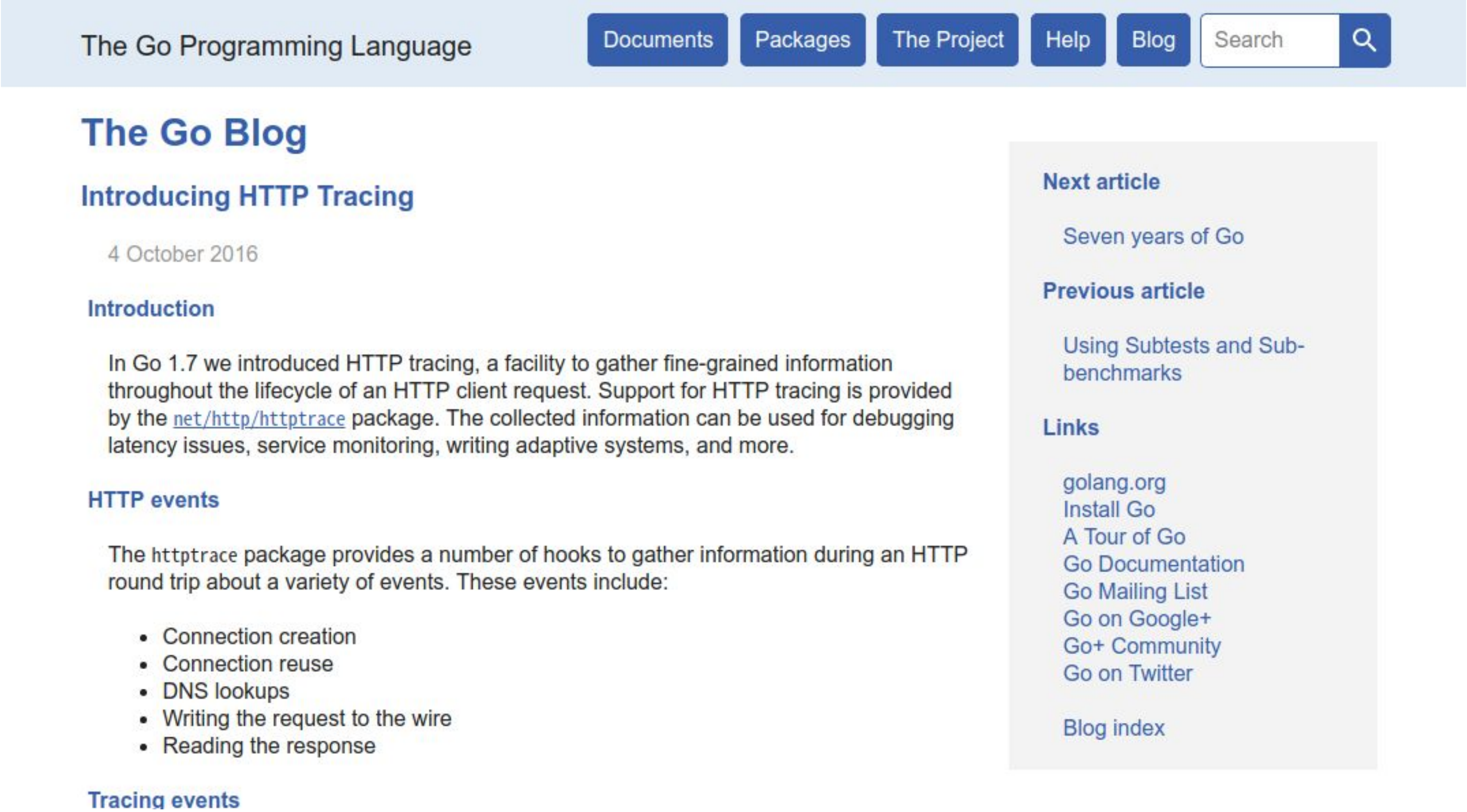
Spotlight on external Go libraries

Links

golang.org
Install Go
A Tour of Go
Go Documentation
Go Mailing List
Go on Google+
Go+ Community
Go on Twitter

Google Cloud

blog.golang.org/profiling-go-programs



The Go Blog

Introducing HTTP Tracing

4 October 2016

Introduction

In Go 1.7 we introduced HTTP tracing, a facility to gather fine-grained information throughout the lifecycle of an HTTP client request. Support for HTTP tracing is provided by the `net/http/httptrace` package. The collected information can be used for debugging latency issues, service monitoring, writing adaptive systems, and more.

HTTP events

The `httptrace` package provides a number of hooks to gather information during an HTTP round trip about a variety of events. These events include:

- Connection creation
- Connection reuse
- DNS lookups
- Writing the request to the wire
- Reading the response

Tracing events

Google Cloud

blog.golang.org/http-tracing

Definitions

Tracing

Stream of the event logs

- Latency
- Timing chart of goroutines

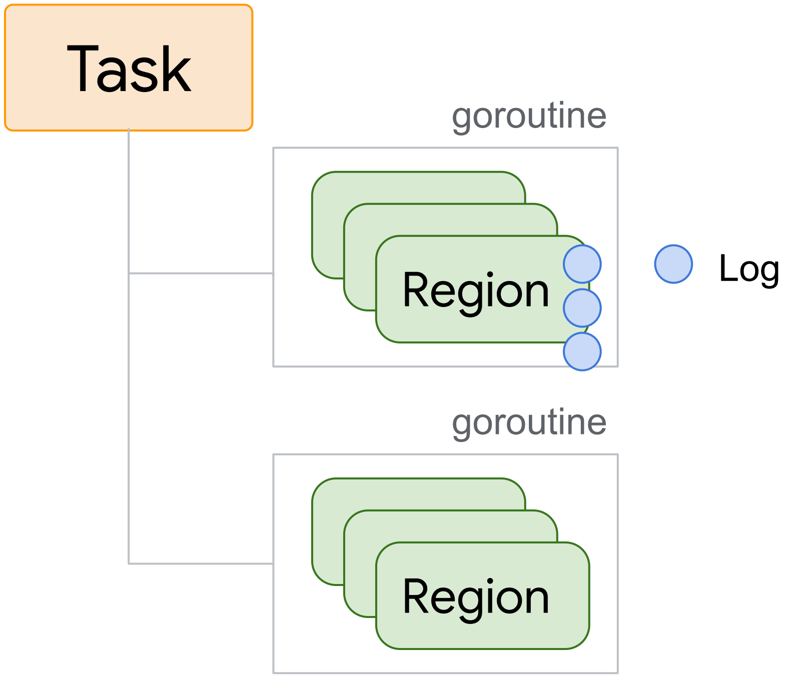
Profiling

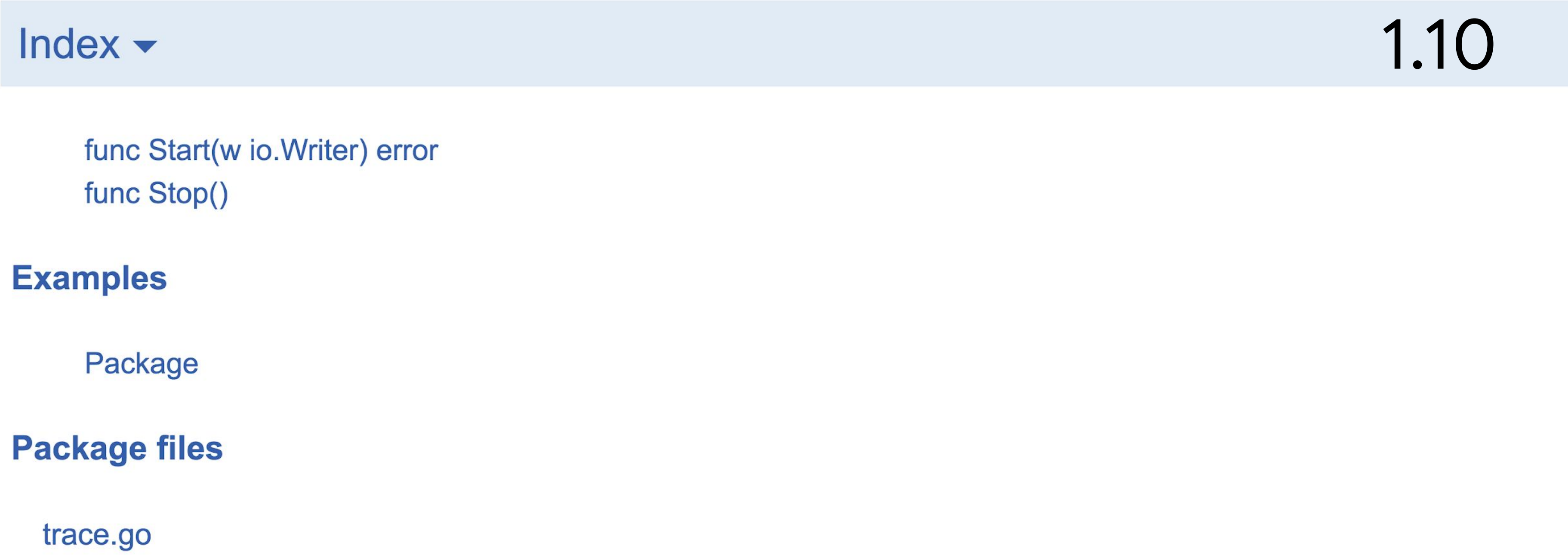
Statistical summary of the sampled events.

- CPU usage
- # of goroutines

What's new of Tracing in Go 1.11

- User annotation
  - trace.Task
  - trace.Region
  - trace.Log





Index ▾

1.10

func Start(w io.Writer) error

func Stop()

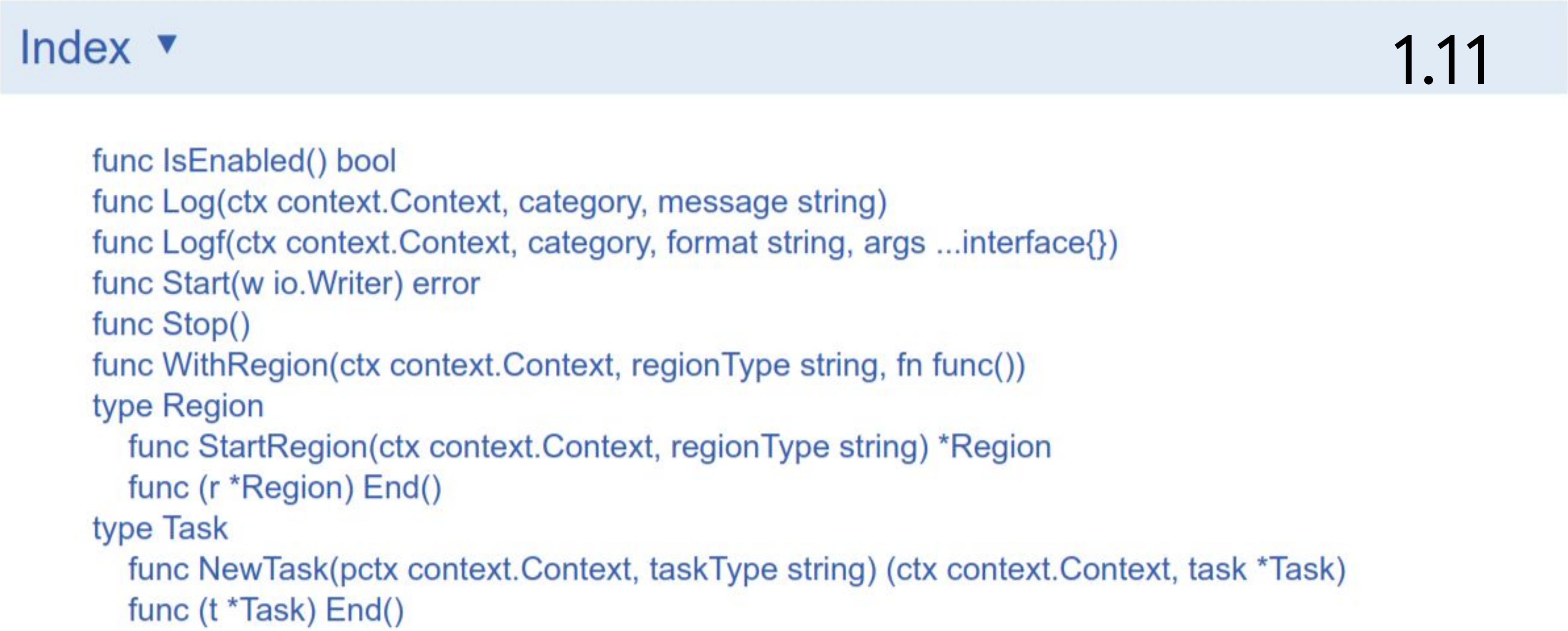
Examples

Package

Package files

trace.go

golang.org/pkg/runtime/trace/

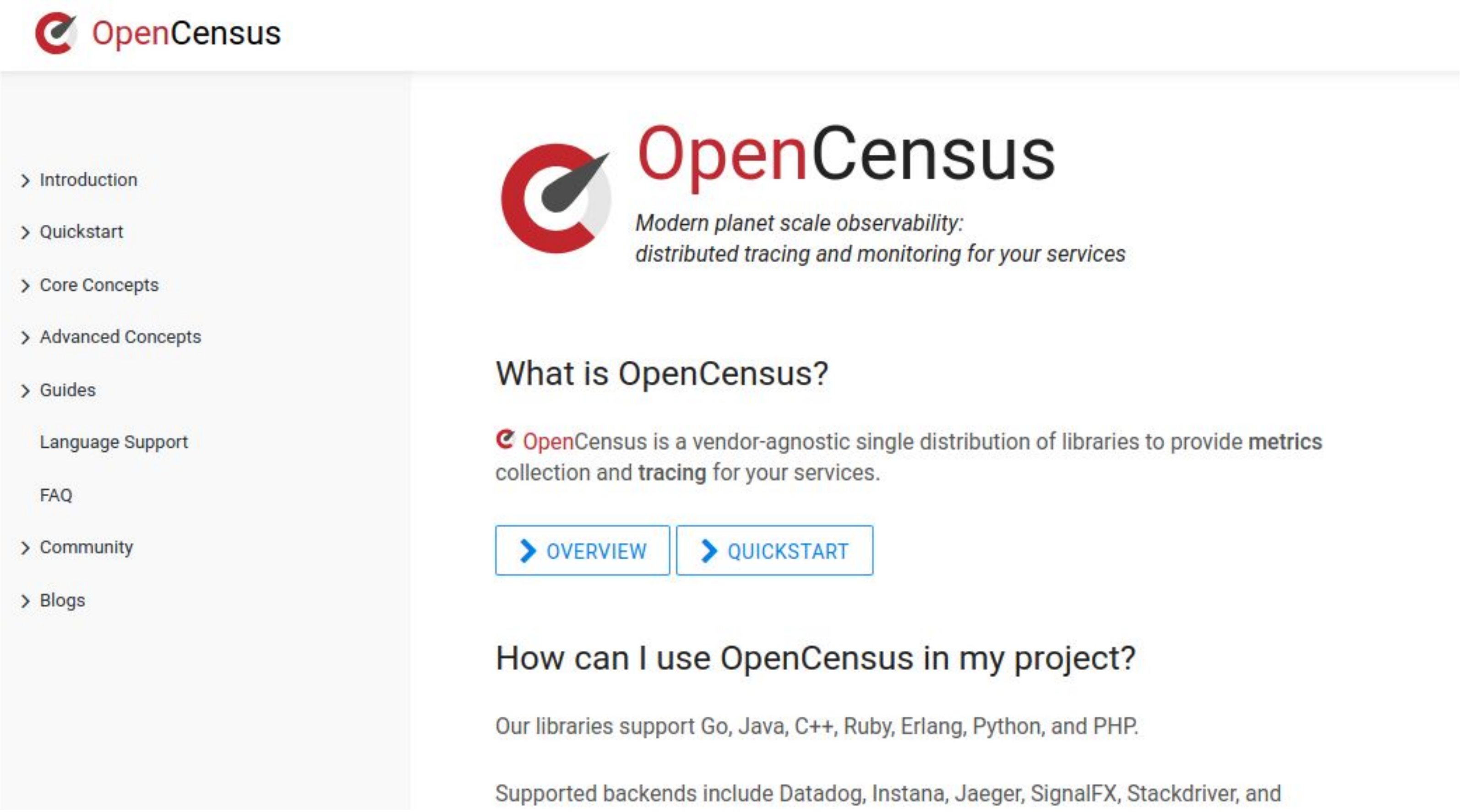


Index ▼

1.11

```
func IsEnabled() bool
func Log(ctx context.Context, category, message string)
func Logf(ctx context.Context, category, format string, args ...interface{})
func Start(w io.Writer) error
func Stop()
func WithRegion(ctx context.Context, regionType string, fn func())
type Region
    func StartRegion(ctx context.Context, regionType string) *Region
    func (r *Region) End()
type Task
    func NewTask(pctx context.Context, taskType string) (ctx context.Context, task *Task)
    func (t *Task) End()
```

tip.golang.org/pkg/runtime/trace/



OpenCensus

Modern planet scale observability:
distributed tracing and monitoring for your services

What is OpenCensus?

OpenCensus is a vendor-agnostic single distribution of libraries to provide **metrics** collection and **tracing** for your services.

How can I use OpenCensus in my project?

Our libraries support Go, Java, C++, Ruby, Erlang, Python, and PHP.

Supported backends include Datadog, Instana, Jaeger, SignalFX, Stackdriver, and

opencensus.io

Partners & Contributors

Google

DATADOG

orijtech

SignalFx

C.e.S.A.R

THE CREATIVE FEW

Microsoft

JAEGER

ZIPKIN

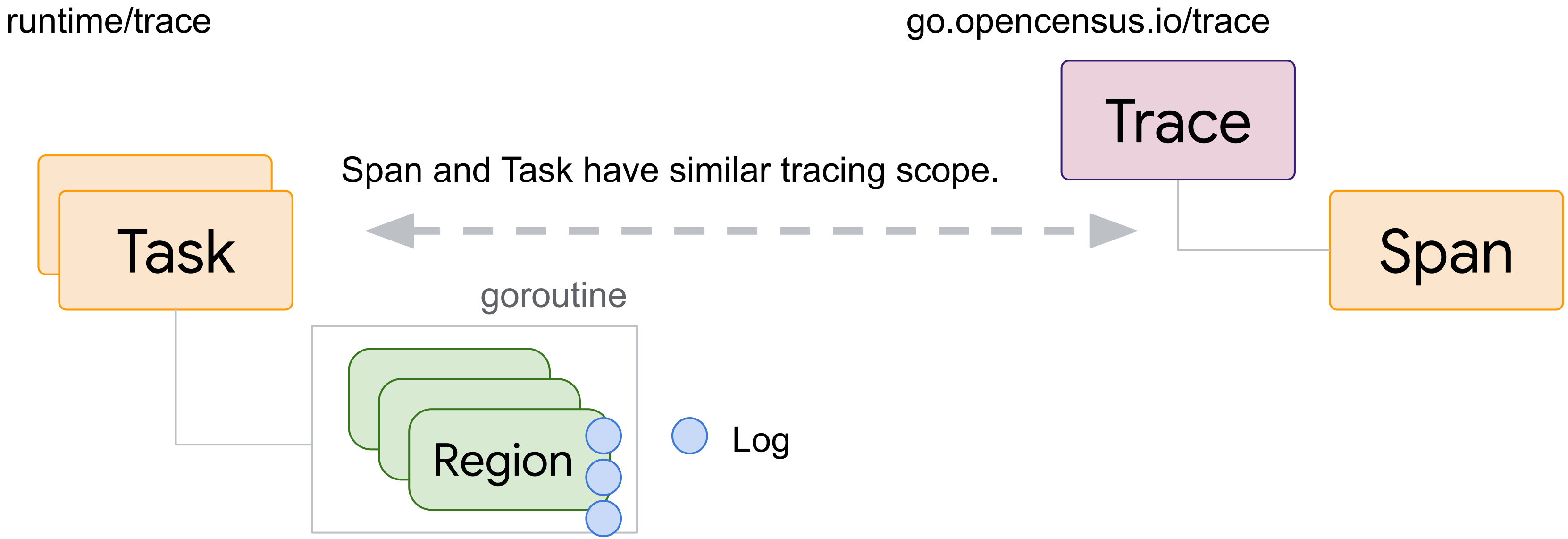
solarwinds

Stackdriver

Prometheus

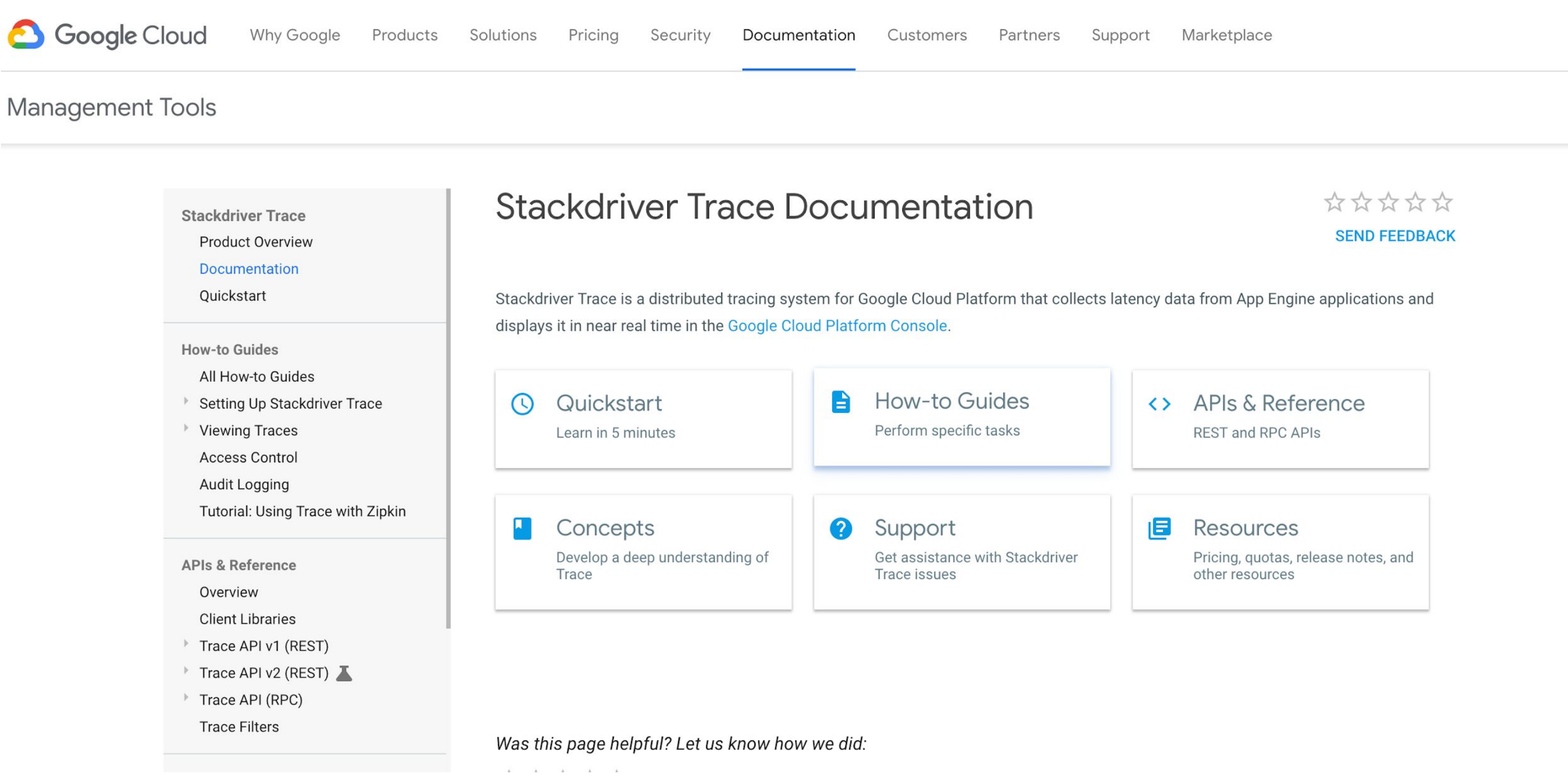
INSTANA

runtime/trace and go.opencensus.io/trace



The output format of the execution tracer is hard to parse, and go tool trace is the only canonical tool that can understand this format. There is no easy way to automatically attaching execution tracer data to the distributed traces—hence we collect them separately and correlate later.

@rakyll
Developer Advocate, Google



Stackdriver Trace Documentation

Stackdriver Trace is a distributed tracing system for Google Cloud Platform that collects latency data from App Engine applications and displays it in near real time in the Google Cloud Platform Console.

Quickstart
Learn in 5 minutes

How-to Guides
Perform specific tasks

APIs & Reference
REST and RPC APIs

Concepts
Develop a deep understanding of Trace

Support
Get assistance with Stackdriver Trace issues

Resources
Pricing, quotas, release notes, and other resources

Was this page helpful? Let us know how we did:

Google Cloud

cloud.google.com/trace/docs/

Demo

Demo scenario

- Try runtime/trace in 1.10
- Try runtime/trace in 1.11rc1 with user annotation
- Use Open Census and confirm *span* corresponds to *runtime/trace.Task*
- Understand contrib exporter is easily linked to *span* (eg. Stackdriver)

Appendix

Blog post/Presentation

Debugging Latency in Go 1.11

opencensus.io

golang.org/pkg/runtime/trace/

Thank you

Google Cloud