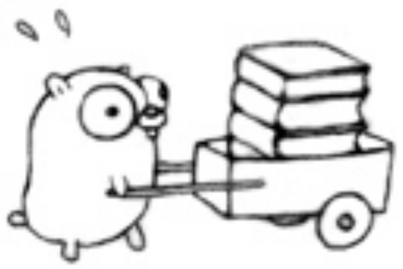
CLOUDFLARE



Serialization in Go

Albert Strasheim
Systems Gopher

About CloudFlare and me

- ➔ Content Delivery Network
- ➔ DNS, Railgun built with Go
- ➔ For some reason, people are trying to send us all their NTP packets...
- ➔ Working on log collection, processing and analytics, mostly in Go

Problem statement

- Understand your distributed system
- Generate, serialize, compress, send, decompress, deserialize, process, store and reprocess events
- This talk: inspired by request log events
  - A few hundred bytes
  - 10-100 fields, mixed bag of field types

Log Schema

```
message Log {
    optional sfixed64 timestamp = 1;
    optional uint32 zoneId = 2;
    optional ZonePlan zonePlan = 3;
    optional CacheStatus cacheStatus = 7;
    optional bytes serverIp = 8;
    optional uint64 bytesDlv = 11;
    optional string rayId = 12;
}
```

Overview

- ➔ encoding/json
  - ◆ Friends: encoding/gob, encoding/xml
- ➔ goprotobuf
- ➔ gogoprotobuf
- ➔ go-capnproto
- ➔ Benchmarks

Standard library packages

encoding/json and friends

Standard library packages

- Go code is your schema
- Uses reflection
- Struct tags can customize behaviour
- You could write your own
- Use Go as your schema language for *anything* with the help of go/parser

encoding/json: Schema

```
type Log struct {
    Timestamp  int64    `json:"timestamp"`
    ZoneId     uint32   `json:"zoneId"`
    ZonePlan   ZonePlan `json:"omitempty"`
    ServerName string   `json:"-"`
}
```

encoding/json: API

```
func Marshal(v interface{}) ([]byte, error)

func MarshalIndent(v interface{}, prefix,
    indent string) ([]byte, error)

func Unmarshal(data []byte,
                   v interface{}) error
```

encoding/json: API

```
func NewDecoder(r io.Reader) *Decoder
func (dec *Decoder)
  Decode(v interface{}) error

func NewEncoder(w io.Writer) *Encoder
func (enc *Encoder)
  Encode(v interface{}) error
```

encoding/json: API

```
type Marshaler interface {
  MarshalJSON() ([]byte, error)
}
type Unmarshaler interface {
  UnmarshalJSON([]byte) error
}
```

encoding/gob

- Easy for Go-to-Go communication
- Supports more types
- Self-describing stream
- Could be implemented in other languages

encoding/xml

It exists. Good luck!

goprotobuf

Protocol Buffers

Protocol Buffers

- Series of key-value pairs
- Key is a field number and wire type
    - Variable-length integers
    - Length-delimited fields like strings
    - Fixed-length 32-bit and 64-bit values
- ProtoText
- Lots of libraries, lots of languages

Protocol Buffers: Schema

```
message Log {
    optional sfixed64 timestamp = 1;
    optional uint32 zoneId = 2;
    optional ZonePlan zonePlan = 3;
    repeated bytes serverIps = 8;
    optional uint64 bytesDlv = 11;
    optional string rayId = 12;
}
enum ZonePlan { FREE = 1; PRO = 2; }
```

goprotobuf: Generated code

```
type Log struct {
    Timestamp *int64
        `protobuf:"fixed64,1,opt,name=timestamp"
         json:"timestamp,omitempty"`
    ZoneId *uint32
        `protobuf:"varint,2,opt,name=zoneId" …`
    XXX_unrecognized []byte `json:"-"` }
```

goprotobuf: API

```
type Message interface {
    Reset()
    String() string
    ProtoMessage()
}
```

goprotobuf: API

```
func Marshal(pb Message) ([]byte, error)

func Unmarshal(buf []byte, pb Message) error

type Buffer struct
func (p *Buffer) Marshal(pb Message) error
func (p *Buffer) Unmarshal(pb Message) error
```

CLOUDFLARE

gogoprotobuf

goprotobuf with gadgets

gogoprotobuf: Extensions

- nullable

  `Timestamp *int64 -> int64`

- embed: embed field in struct

  `type Log struct { LogPart }`

- customtype

  `[]byte/string` -> something else

gogoprotobuf: More extensions

```
type Marshaler interface
type Unmarshaler interface
```

- ➔ Generate fast Marshal method
- ➔ Generate fast Unmarshal method

gogoprotobuf: More extensions

- stringer: generate String()
- equal: generate Equal()
- testgen: generate marshalling tests
- benchgen: generate benchmarks
- and lots more...

gogoprotobuf: Schema changes

```
import "code.google.com/p/gogoprotobuf/…
    gogoproto/gogo.proto";

option (gogoproto.testgen_all) = true;

message Log {
    option (gogoproto.nullable) = false;
    optional bytes ip = 8 [(gogoproto.customtype)="IP"];
}
```

gogoprotobuf: code generation

```
protoc --gogo_out=...
  /path/to/some.proto

code.google.com/p/gogoprotobuf/…
  protoc-gen-gogo/main.go
```

Katydid

- Tree automata
- Write queries to match messages in a stream of protocol buffers
- http://arborist.katydid.ws/
- github.com/awalterschulze/katydid

CLOUDFLARE

CAP'N PROTO

cerealization protocol

infinitely faster!

CLOUDFLARE

Cap’n Proto

- ➔ Machine-friendly format
- ➔ No explicit serialization step
- ➔ Packing to reduce zeros
- ➔ RPC with promise pipelining
- ➔ See also: sandstorm.io

CLOUDFLARE

Cap’n Proto: Schema

```
@0xe68600092ff3f0a59;

using Go = import "go.capnp";
$Go.package("mypkg");

struct Log { … }
```

Cap’n Proto: Schema

```
struct Log {
   timestamp @0 :Int64;
   zoneId @1 :UInt32;
   zonePlan @2 :ZonePlan;
   http @3 :HTTP;
   remoteIp @4 :Data;
   rayId @5 :Text; }
```

CLOUDFLARE

Cap’n Proto: Annotations

```
enum ZonePlan {
  free @1 $Go.tag("Free");
  pro @2 $Go.tag("Pro");
  biz @3 $Go.tag("Business");
  ent @4 $Go.tag("Enterprise");
}
```

CLOUDFLARE

go-capnproto: capnpc-go

```
capnp compile -ogo log.capnp
```

Parses log.capnp and sends it as a CodeGeneratorRequest to capnpc-go on standard input

capnp: more fun

```
capnp compile -o /bin/cat log.capnp |
  capnp decode schema.capnp
    CodeGeneratorRequest
```

go-capnproto: Generated code

```
import C "github.com/jmckaskill/go-capnproto"

type Log C.Struct

func NewRootLog(s *C.Segment) Log
func NewLog(s *C.Segment) Log
func ReadRootLog(s *C.Segment) Log
func NewLogList(s *C.Segment, sz int) Log_List
```

go-capnproto: Generated code

```
func (s Log) Timestamp() int64 {
    return int64(C.Struct(s).Get64(0)) }
func (s Log) SetTimestamp(v int64){
    C.Struct(s).Set64(0, uint64(v)) }
func (s Log) RayId() string {
    return C.Struct(s).GetObject(5).ToText() }
func (s Log) SetRayId(v string) {
    C.Struct(s).SetObject(5, s.Segment.NewText(v)) }
```

go-capnproto: API

```
b := make([]byte, 0, 16<<10)
segment := capn.NewBuffer(b)
event := capnp.NewRootLog(segment)
event.SetTimestamp(…)
err := segment.WriteTo(writer)
```

go-capnproto: API

```
var buf bytes.Buffer
for {
   seg, err :=
      capn.ReadFromStream(reader, &buf)
   event := ReadRootLog(seg)
}
```

go-capnproto: Next Steps

- Fix a few minor bugs
- Optimize
- API tweaks
- Split code generation into a library
- Wrap the C++ parser with SWIG

Benchmarks

My attempt to give you real numbers

Benchmarks: Populate struct fields

```
BenchmarkPopulatePb
        1180 ns/op          16 allocs/op
BenchmarkPopulateGogopb
        390 ns/op           3 allocs/op
BenchmarkPopulateCapnp
        4509 ns/op          2 allocs/op
```

Benchmarks: Serialization

```
BenchmarkMarshalJSON     11918 ns/op
BenchmarkMarshalPb       2487 ns/op
BenchmarkMarshalGogopb   581 ns/op
BenchmarkMarshalCapnp    183 ns/op
```

CLOUDFLARE

Benchmarks: Deserialization

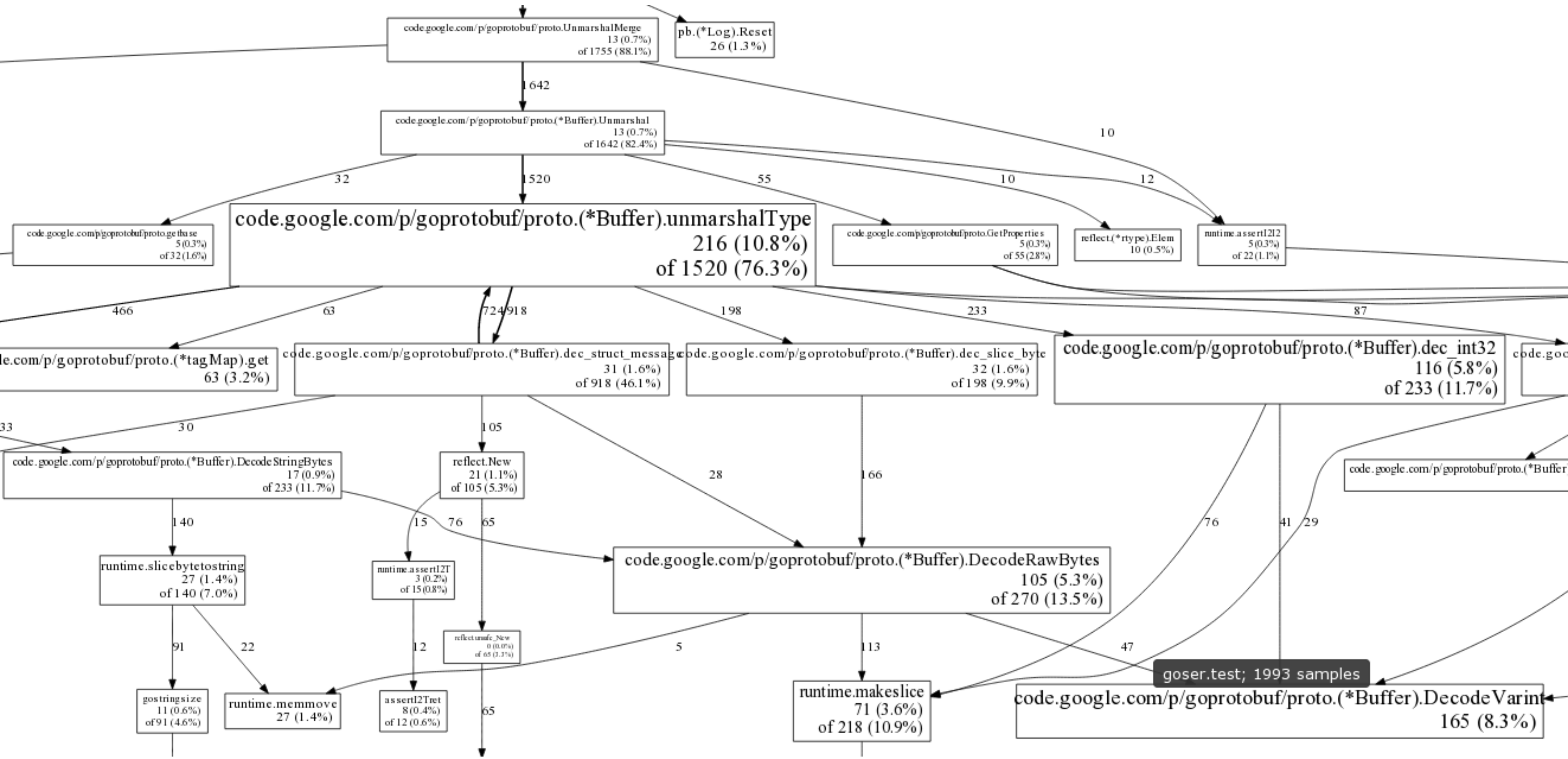
```
BenchmarkUnmarshalJSON              31174 ns/op
BenchmarkUnmarshalPb                3262 ns/op
BenchmarkUnmarshalGogopb            894 ns/op
BenchmarkUnmarshalCapnp             879 ns/op
BenchmarkUnmarshalCapnpZeroCopy     467 ns/op
```

CLOUDFLARE

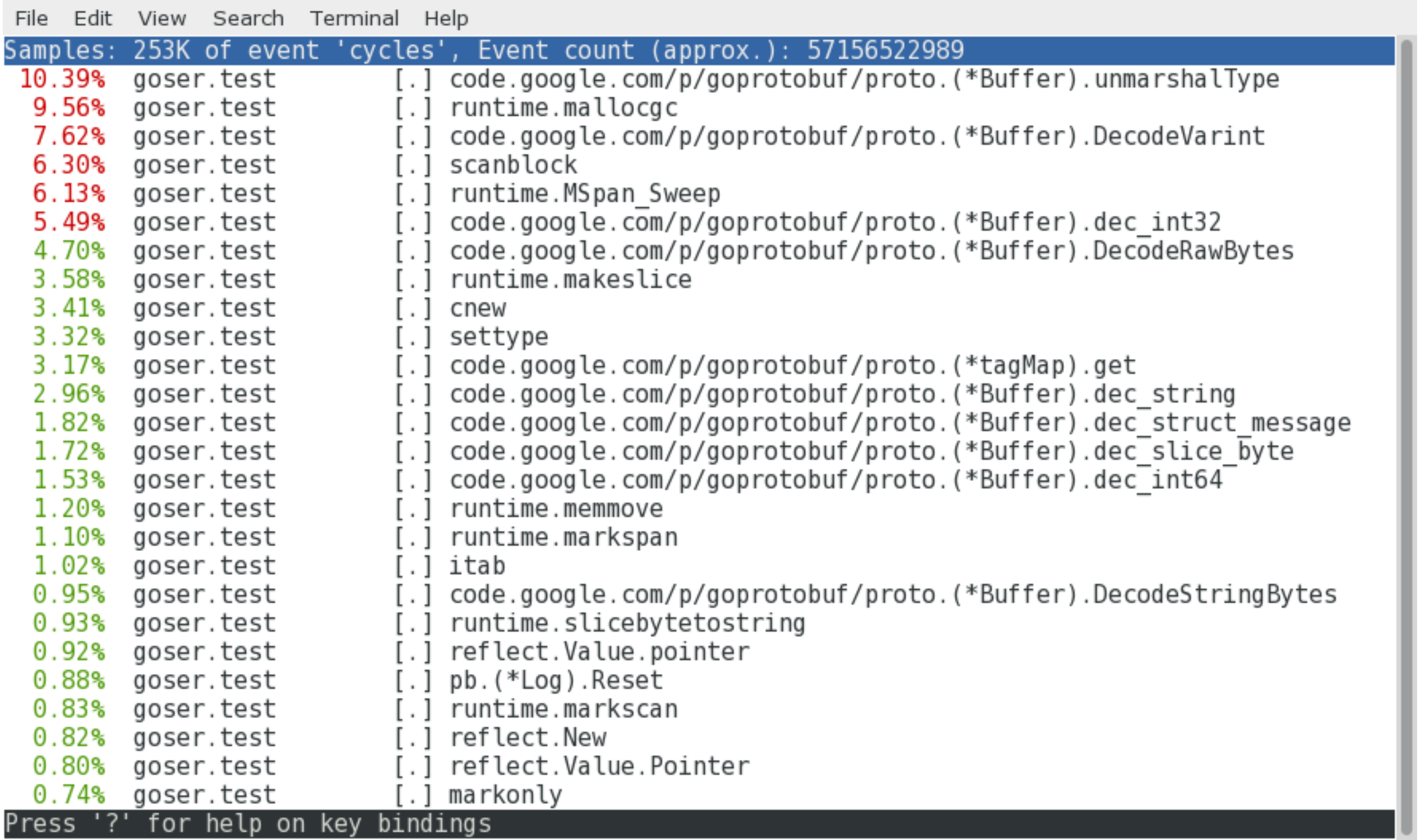
Profiling tools

```
go test -benchmem
go test -cpuprofile=cpu.prof
go tool pprof
```

perf top on Linux


code.google.com/p/goprotobuf/proto.UnmarshalMerge 13 (0.7%) of 1755 (88.1%)
pb.(*Log).Reset 26 (1.3%)
1642
code.google.com/p/goprotobuf/proto.(*Buffer).Unmarshal 13 (0.7%) of 1642 (82.4%)
10
32
1520
55
10
12
code.google.com/p/goprotobuf/proto.getbase 5 (0.3%) of 32 (1.6%)
code.google.com/p/goprotobuf/proto.(*Buffer).unmarshalType 216 (10.8%) of 1520 (76.3%)
code.google.com/p/goprotobuf/proto.GetProperties 5 (0.3%) of 55 (2.8%)
reflect.(*rtype).Elem 10 (0.5%)
runtime.assertI2I2 5 (0.3%) of 22 (1.1%)
466
63
724
918
198
233
87
le.com/p/goprotobuf/proto.(*tagMap).get 63 (3.2%)
code.google.com/p/goprotobuf/proto.(*Buffer).dec_struct_message 31 (1.6%) of 918 (46.1%)
code.google.com/p/goprotobuf/proto.(*Buffer).dec_slice_byte 32 (1.6%) of 198 (9.9%)
code.google.com/p/goprotobuf/proto.(*Buffer).dec_int32 116 (5.8%) of 233 (11.7%)
code.goo
33
30
105
code.google.com/p/goprotobuf/proto.(*Buffer).DecodeStringBytes 17 (0.9%) of 233 (11.7%)
reflect.New 21 (1.1%) of 105 (5.3%)
28
166
code.google.com/p/goprotobuf/proto.(*Buffer
140
15
76
65
76
41
29
runtime.slicebytetostring 27 (1.4%) of 140 (7.0%)
runtime.assertI2T 3 (0.2%) of 15 (0.8%)
code.google.com/p/goprotobuf/proto.(*Buffer).DecodeRawBytes 105 (5.3%) of 270 (13.5%)
reflect.unsafe_New 0 (0.0%) of 65 (3.3%)
91
22
12
5
113
47
goser.test; 1993 samples
gostringsize 11 (0.6%) of 91 (4.6%)
runtime.memmove 27 (1.4%)
assertI2Tret 8 (0.4%) of 12 (0.6%)
65
runtime.makeslice 71 (3.6%) of 218 (10.9%)
code.google.com/p/goprotobuf/proto.(*Buffer).DecodeVarint 165 (8.3%)

CLOUDFLARE



```
Samples: 253K of event 'cycles', Event count (approx.): 57156522989
 10.39%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).unmarshalType
  9.56%  goser.test          [.] runtime.mallocgc
  7.62%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).DecodeVarint
  6.30%  goser.test          [.] scanblock
  6.13%  goser.test          [.] runtime.MSpan_Sweep
  5.49%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).dec_int32
  4.70%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).DecodeRawBytes
  3.58%  goser.test          [.] runtime.makeslice
  3.41%  goser.test          [.] cnew
  3.32%  goser.test          [.] settype
  3.17%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*tagMap).get
  2.96%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).dec_string
  1.82%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).dec_struct_message
  1.72%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).dec_slice_byte
  1.53%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).dec_int64
  1.20%  goser.test          [.] runtime.memmove
  1.10%  goser.test          [.] runtime.markspan
  1.02%  goser.test          [.] itab
  0.95%  goser.test          [.] code.google.com/p/goprotobuf/proto.(*Buffer).DecodeStringBytes
  0.93%  goser.test          [.] runtime.slicebytetostring
  0.92%  goser.test          [.] reflect.Value.pointer
  0.88%  goser.test          [.] pb.(*Log).Reset
  0.83%  goser.test          [.] runtime.markscan
  0.82%  goser.test          [.] reflect.New
  0.80%  goser.test          [.] reflect.Value.Pointer
  0.74%  goser.test          [.] markonly
Press '?' for help on key bindings
```

Conclusion

Build benchmarks with your own data

Questions?

Slides on the CloudFlare blog soon

Code at github.com/cloudflare/goser