Sample Your Traffic

But Keep the Good Stuff!

Ben Hartshorne
ben@honeycomb.io
@maplebed

https://speakerdeck.com/maplebed/sample-your-traffic

usenix
LISA17

Samples? Is that like Swag?

- Who am I?
- What's Sampling?
- Methods for Choosing Samples
- In Practice

honeycomb.io

- My Employer
- Observability SaaS
- Powertool for Engineers
- Visualization of events
- Interactive, Exploratory



Ben Hartshorne

- Opsen turned Engineer
- Spent too long in Ganglia
- Linden Lab, Wikimedia, Parse, Facebook
- Really digs pretty graphs
- Finally building tools I’ve always wanted to have

ben@honeycomb.io

@maplebed

What is Sampling?

Selecting

a subset of a group

in a way that

it represents the group

What is Sampling?

Hang on,

Why would we do this?

Isn't it inaccurate?

Don't we want all the data?

What is Sampling?

How do we reduce data?

- Measure fewer things
- Send aggregates
- Send samples

What is Sampling?

A **subset of a group** that **represents the group**

Timestamp → UTC	BuildID	GojiPattern	Method	ResponseContentLength	ResponseHttpStatus	ResponseTime_ms
2017-10-24 14:33:51.137	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	695	200	37
2017-10-24 14:34:27.570	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:36:28.793	"5162"	"/:team_slug/datasets/:slug/triggers"	"GET"	8,514	200	32
2017-10-24 14:36:31.568	"5162"	"/user_event/:event_type"	"POST"	3	200	325
2017-10-24 14:37:15.526	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	36	200	7
2017-10-24 14:37:15.773	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	480,804	200	209
2017-10-24 14:39:24.155	"5162"	"/user_event/:event_type"	"POST"	3	200	259
2017-10-24 14:39:24.242	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	12,277	200	91
2017-10-24 14:39:44.222	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/json"	"GET"	691	200	5
2017-10-24 14:39:45.612	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:39:48.359	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:40:11.097	"5162"	"/:team_slug/datasets/:slug"	"GET"	12,649	200	353
2017-10-24 14:40:13.788	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:40:34.229	"5162"	"/user_event/:event_type"	"POST"	3	200	154
2017-10-24 14:40:34.395	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	39,145	200	56
2017-10-24 14:40:48.524	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	3,211	200	8
2017-10-24 14:40:49.668	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:41:05.125	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	3,211	200	14
2017-10-24 14:41:05.351	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:41:06.858	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5

What is Sampling?

A **subset of a group** that **represents the group:** Sample Rate **1/5** or **20%**

Timestamp ⌄ UTC	BuildID	GojiPattern	Method	ResponseContentLength	ResponseHttpStatus	ResponseTime_ms
2017-10-24 14:33:51.137	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	695	200	37
2017-10-24 14:34:27.570	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:36:28.793	"5162"	"/:team_slug/datasets/:slug/triggers"	"GET"	8,514	200	32
2017-10-24 14:36:31.568	"5162"	"/user_event/:event_type"	"POST"	3	200	325
2017-10-24 14:37:15.526	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	36	200	7
2017-10-24 14:37:15.773	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	480,804	200	209
2017-10-24 14:39:24.155	"5162"	"/user_event/:event_type"	"POST"	3	200	259
2017-10-24 14:39:24.242	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	12,277	200	91
2017-10-24 14:39:44.222	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/json"	"GET"	691	200	5
2017-10-24 14:39:45.612	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:39:48.359	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:40:11.097	"5162"	"/:team_slug/datasets/:slug"	"GET"	12,649	200	353
2017-10-24 14:40:13.788	"5162"	"/account"	"GET"	29	302	0
2017-10-24 14:40:34.229	"5162"	"/user_event/:event_type"	"POST"	3	200	154
2017-10-24 14:40:34.395	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	39,145	200	56
2017-10-24 14:40:48.524	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	3,211	200	8
2017-10-24 14:40:49.668	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:41:05.125	"5162"	"/:team_slug/datasets/:slug/markers/:start_time_unix/:end_time_unix"	"GET"	3,211	200	14
2017-10-24 14:41:05.351	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5
2017-10-24 14:41:06.858	"5162"	"/:team_slug/datasets/:slug/result/:query_run_pk/status"	"GET"	72	200	5

Note: rows chosen by running `echo $(($RANDOM % 20))` 4 times

What is Sampling?

Traffic is not all equal…

- Infrequent vs. frequent
- Writes vs. reads
- Error vs. success
- Business-relevant characteristics

What is Sampling?

Traffic is not all equal…

- Create Sample Keys

Status Code

URL

Service Name

Request ID

Customer ID

Errors

All of the above

What is Sampling?

But the MATH!!!

Value	Sample Rate	Calc Value
3	10	30
5	20	100
3	5	15
10	1	10

Average

(30 + 100 + 15 + 10) /
(10+20+5+1) =
155/36 = 4.3

What is Sampling?

Some Details…

- Measure rates as a ratio
- Choose representative elements randomly
- Communicate your choice with your visualization engine…
  - On a **per-event** basis

Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- Rate Limited
- Dynamic Map
- Remote Source
- Compositions

Want some code?

github.com/honeycombio/dynsampler-go

github.com/jaegertracing/
jaeger-client-go/blob/master/sampler.go

Sampling Algorithms

- **Constant Rate**
- Consistent Sample
- Map of Rates
- Rate Limited
- Dynamic Map
- Remote Source
- Compositions

Constant Rate

Sample all traffic at the same fixed rate. Every event has equal probability it will be reported.
Example: 1/50 sample rate

Good for homogenous traffic

Advantages

- Simple
- Predictable

Downsides

- Inflexible
- Hides rare traffic

Sampling Algorithms

- Constant Rate
- **Consistent Sample**
- Map of Rates
- Rate Limited
- Dynamic Map
- Remote Source
- Compositions

Consistent Sample

A given sample key always gives the same result. Allows multiple messages to all be sampled together.
Example: Trace ID

Good for distributed systems

Advantages

- Simple
- Allows multi-part events

Downsides

- Inflexible
- Requires large key space

Sampling Algorithms

- Constant Rate
- Consistent Sample
- **Map of Rates**
- Rate Limited
- Dynamic Map
- Remote Source
- Compositions

Map of Rates

Create a static map of traffic type to sample rate.
Example: HTTP Status Codes

Good for low-cardinality keys

Advantages

- Easy, Clear
- Works in config

Downsides

- Static
- Annoying to maintain

Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- **Rate Limited**
- Dynamic Map
- Remote Source
- Compositions

Rate Limited

Send a specific number of events per time period. Requires specific shapes of traffic to be useful.
Example: Exception Trackers

Good for when whether something happens is important

Advantages

- Simple
- Low Volume

Downsides

- Overly Specific
- Coarse

Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- Rate Limited
- **Dynamic Map**
- Remote Source
- Compositions

Dynamic Map

Express the relationship between the key and sample rate in code, dynamically adjusting the rate based on usage.
Example: per-customer sampling

Best Choice for most server traffic

Advantages

- Flexible
- Responsive

Downsides

- Complex



Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- Rate Limited
- Dynamic Map
- **Remote Source**
- Compositions

Remote Source

Delegate the decision of sample rate to a separate service.
Example: Coordinate sample rates across multiple services

Allows external factors to influence sampling dynamically

Advantages

- Flexible
- Runtime Updatable

Downsides

- Slow
- Complex



Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- Rate Limited
- Dynamic Map
- Remote Source
- **Compositions**

Compositions

Mix and match! Rate limit some traffic while mapping others. The sky's the limit.
Example: Average sample rate with minimum traffic per key

Pick the best of multiple approaches

Advantages
- Flexible
- Open Ended

Downsides
- Complex
- Hidden effects



Sampling Algorithms

- Constant Rate
- Consistent Sample
- Map of Rates
- Rate Limited
- Dynamic Map
- Remote Source
- Compositions

Plenty More

This has been a selection from

- Honeycomb's dynsampler package
- Jaeger's client sampling

It is by no means complete

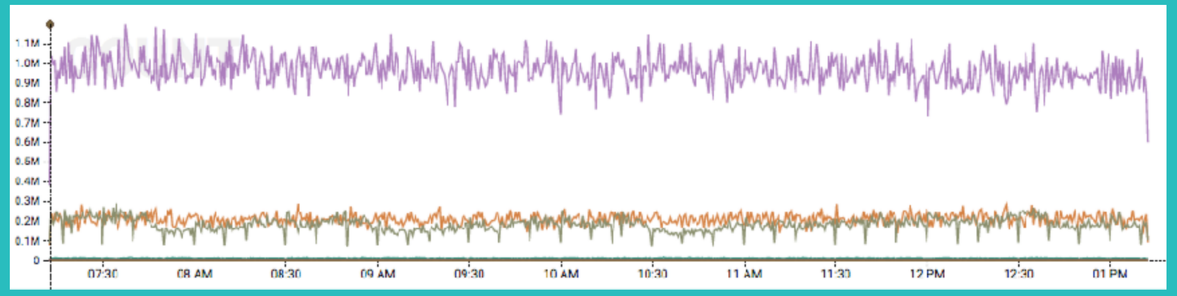
Write some more!
(and contribute them!)

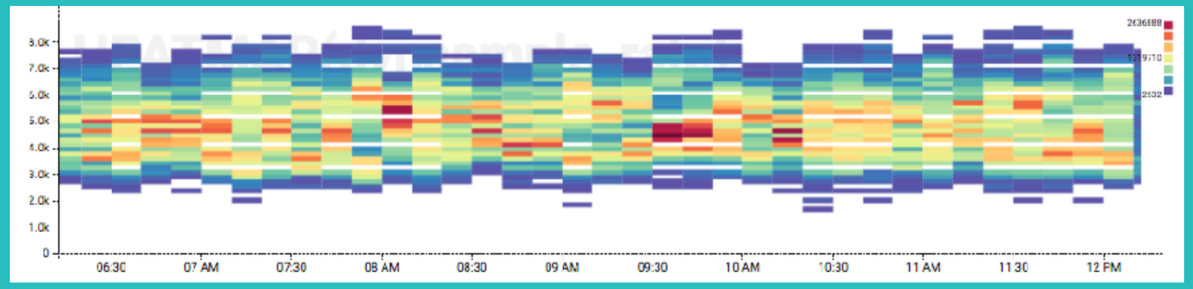
github.com/honeycombio/dynsampler-go

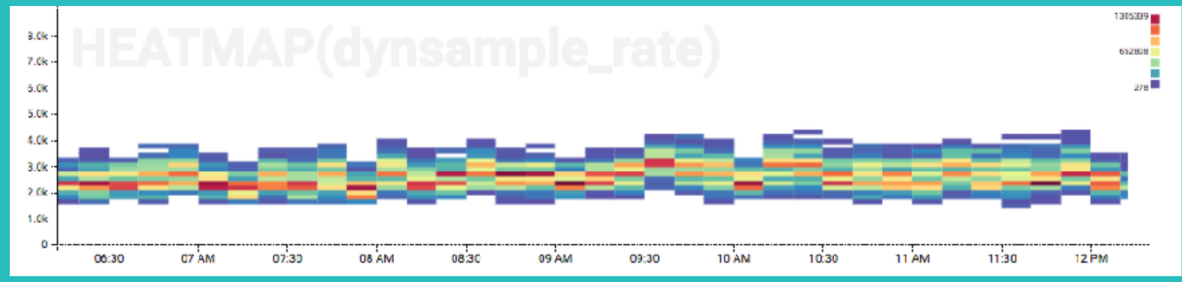
github.com/jaegertracing/
jaeger-client-go/blob/master/sampler.go

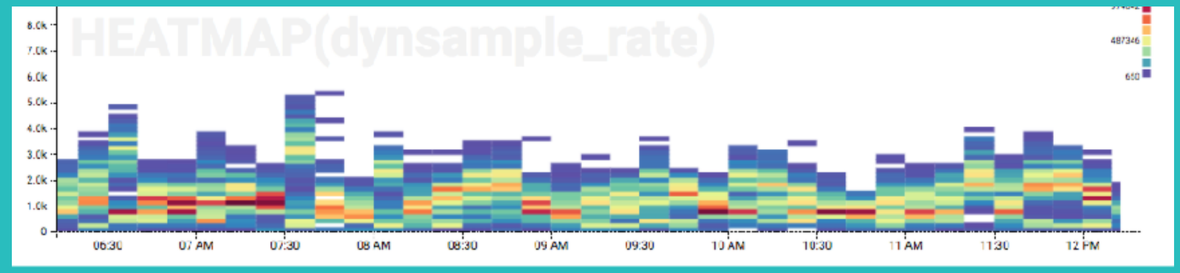
Visualized

- Three traffic volumes
- Each volume is sampled across a different range
- Higher volume has higher sampling
- Extends all the way down to unique keys never getting sampled



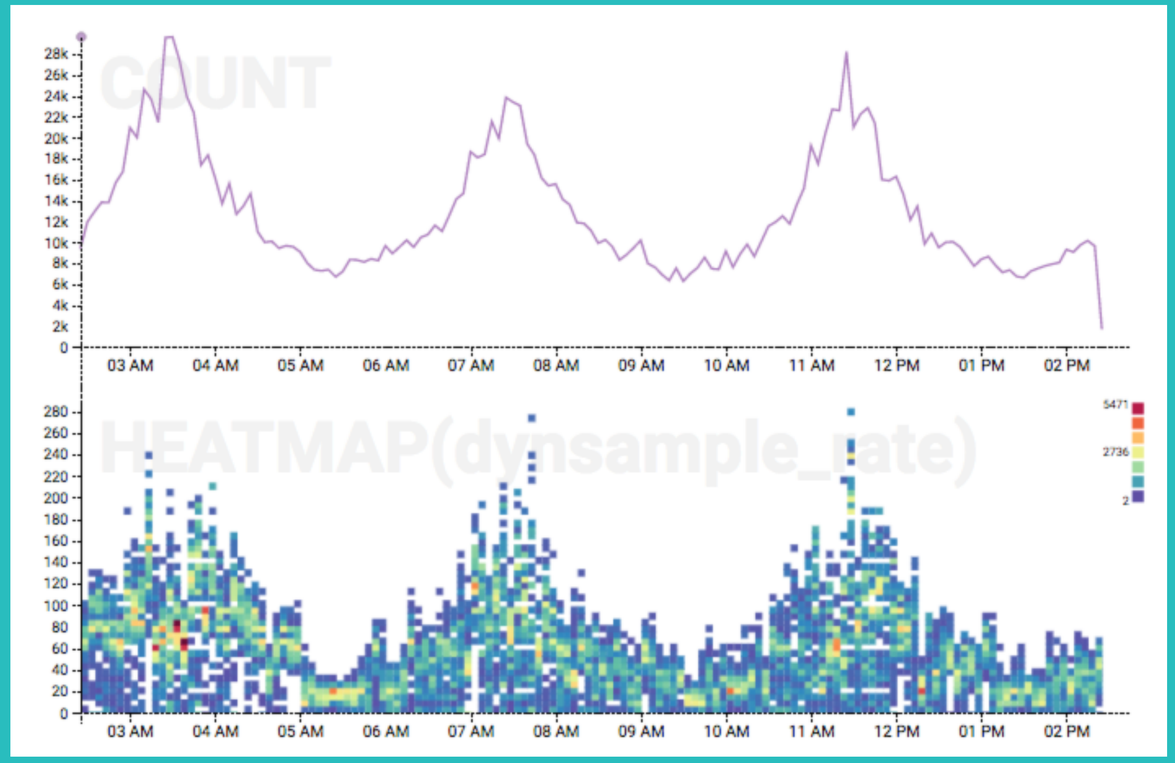




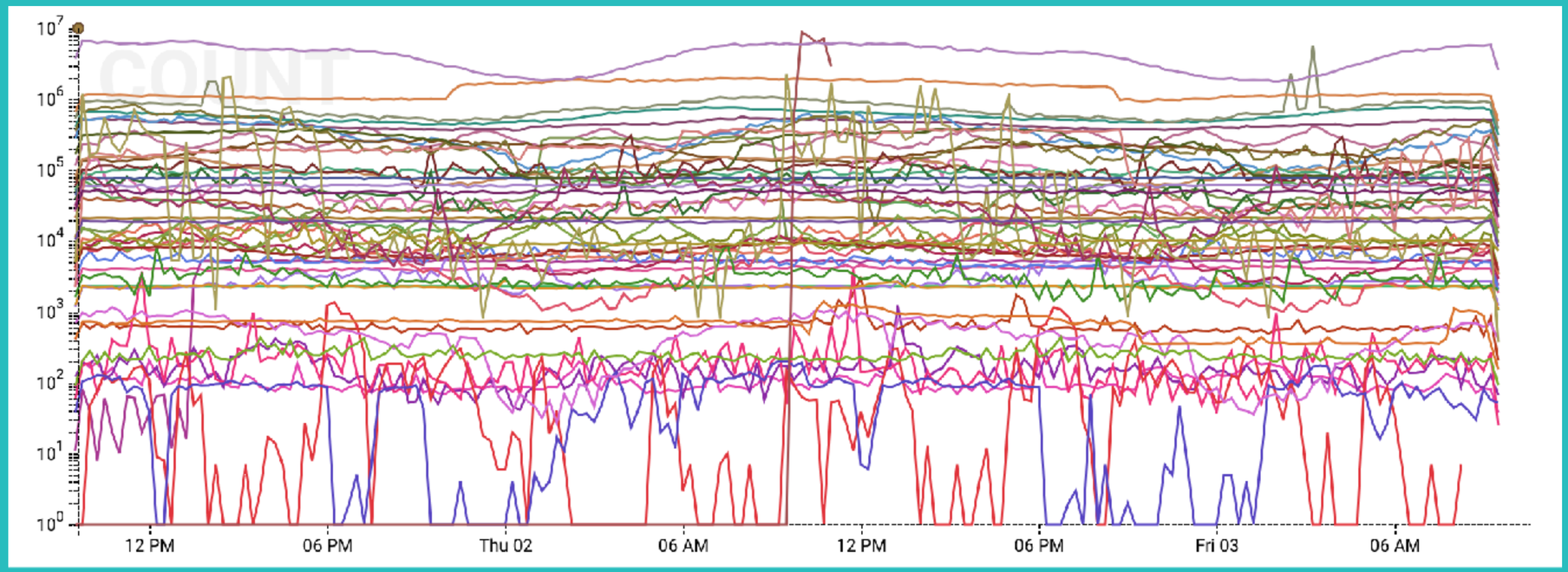


Visualized

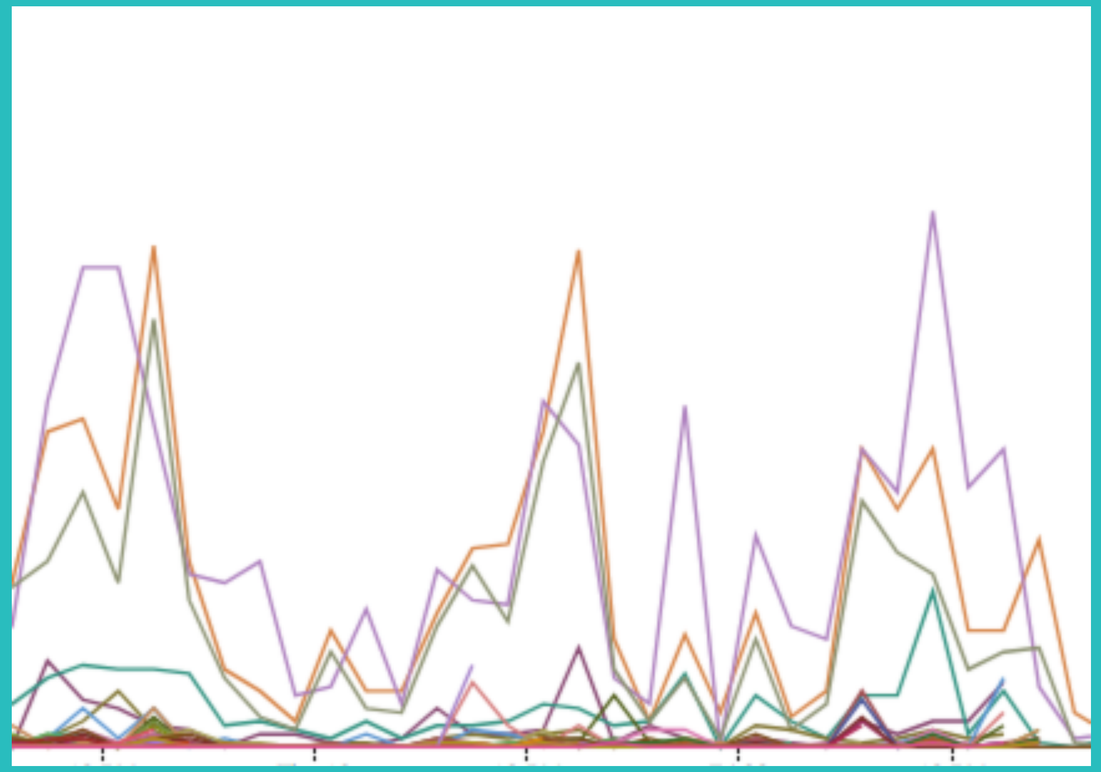
- One traffic source
- Varies over time
- Sample rate adjusts
- Growing or spiking traffic doesn't overwhelm your observability backend



Traffic with a power law distribution makes for the best dynamic sampling



Examples



honeytail

```
if sampler == nil {
    ev.SampleRate = int(options.SampleRate)
} else {
    key := makeDynsampleKey(&ev, options)
    sr := sampler.GetSampleRate(key)
    if rand.Intn(sr) != 0 {
        ev.SampleRate = -1
    } else {
        ev.SampleRate = sr
    }
}
```

- Choose fields to use as the key
- Concatenate those fields with underscores
- Fit a log curve to the frequency of the key

github.com/honeycombio/honeytail



Jaeger

- Sample based on Trace ID and Operation
- Propagate to all spans
- Use Consistent Sampler to sample all spans in a trace or none
- Combine with Rate Limits or Maps to adjust based on traffic

github.com/jaegertracing/jaeger-client-go/



Honeycomb API Server

- A few customers send most of the traffic
- We care about all customers
- Need visibility into low traffic transactions
- Create the dynamic sampling key from:

HTTP Method + HTTP Status + Dataset ID + URL

Honeycomb API Server

HTTP Method + HTTP Status + Dataset ID + URL

- Different datasets are sampled according to volume
- Errors are sampled on a per-dataset basis
- GETs sampled separately from POSTs
- HTTP endpoints sampled independently

Honeycomb API Server

HTTP Method + HTTP Status + Dataset ID + URL

- A new customer makes their first POST
- A normally successful dataset has a few errors
- High volume client errors don't hide infrequent errors
- Infrequent endpoints aren't masked by volume

Conclusion

- Record context-laden events
- Sample them
- Influence your sample rate
- Business and technical goals
- Enable your development teams
- Enable your support teams

Thanks!

ben@honeycomb.io

@maplebed

Slides
speakerdeck.com/maplebed/sample-your-traffic

Try Honeycomb free for 30 days!
honeycomb.io

