Sample Your Traffic

But Keep the Good Stuff!

Ben Hartshorne
ben@honeycomb.io
@maplebed

https://speakerdeck.com/maplebed/sample-your-traffic
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
What is Sampling?

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

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

<table>
<thead>
<tr>
<th>Status Code</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>Request ID</td>
</tr>
<tr>
<td>Customer ID</td>
<td></td>
</tr>
<tr>
<td>Errors</td>
<td>All of the above</td>
</tr>
</tbody>
</table>
What is Sampling?

But the MATH!!!

<table>
<thead>
<tr>
<th>Value</th>
<th>Sample Rate</th>
<th>Calc Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Average

\[
\frac{(30 + 100 + 15 + 10)}{(10+20+5+1)} = 155/36 = 4.3
\]
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
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

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

[GitHub Link](https://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

[Link to GitHub]
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