应用性能监控

最佳实践

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

最佳实践
使用 Jaeger 进行客户端采样配置
使用 Skywalking 进行 Java 应用采集配置

最佳实践
使用 Jaeger 进行客户端采样配置

最近更新时间：2024-04-02 10:09:04

本文将为您介绍如何使用 Jaeger 进行客户端采样配置。

操作背景

在访问量较大时，全链路数据上报可能会导致使用应用性能监控的成本较高。在访问量级较大的情况下，往往会进行数据采样。

说明：

采样指从全量采集的所有链路数据中，采集部分数据进行分析，减少上报数量和链路存储费用，降低使用应用性能监控的成本。

采样解析

以一个简单的调用关系为例：A > B > C（A 服务调用 B 服务，同时，B 服务调用 C 服务）。如果 A 服务被调用的时候没有收到任何追踪信息，A 服务的 Jaeger 库就会创建一个新的追踪(Trace)，分配一个 Trace ID，并根据采样配置决策是否需要保存这次追踪。采样配置决策与请求一起传递到 B 服务和 C 服务，因此我们只需要在 A 服务进行采样配置即可。

操作步骤

采样策略

Jaeger客户端支持4种采样策略，分别是：

Constant (sampler.type=const) ：采样率的可设置的值为 0 和 1，分别表示关闭采样和全部采样。

Probabilistic (sampler.type=probabilistic) ：按照概率采样，取值可在 0 至 1 之间，例如：设置为 0.5 ，表示对 50% 的请求采样。

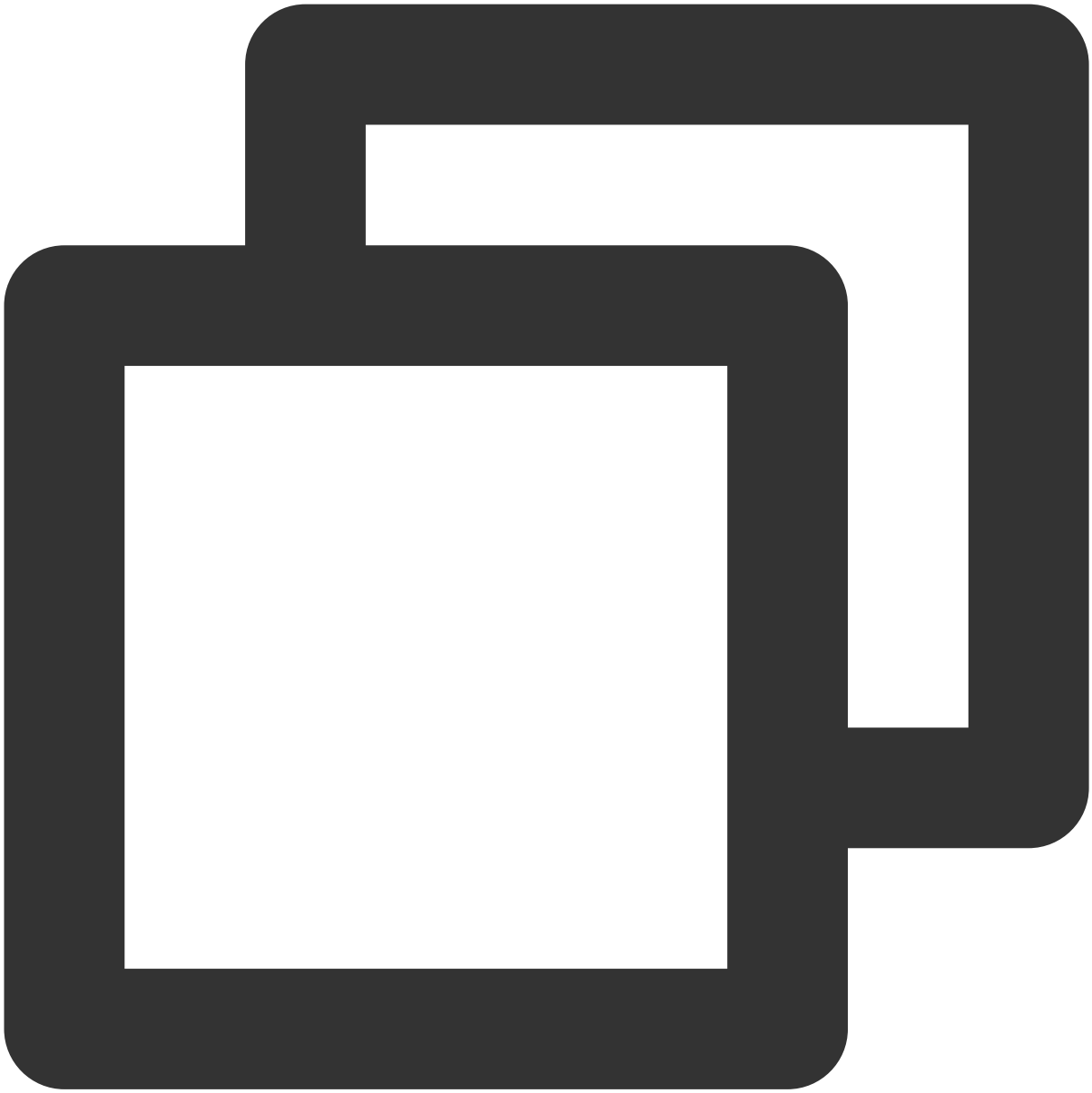
Rate Limiting (sampler.type=ratelimiting)：设置每秒的采样次数上限 。 例如当 `sampler.param = 2.0` 时，它将以每秒 2条链路对请求进行采样。

Remote (sampler.type=remote)：默认策略。 采样遵循远程设置，取值的含义和 probabilistic 相同，都意为采样的概率，但是设置为 remote 后，Client 会从 Jaeger Agent 中动态获取采样率设置。 为了最大程度地减少开销，**Jaeger 默认采用 0.1% 的采样策略采集数据 (1000次里面采集1次)**。

版权所有：腾讯云计算（北京）有限责任公司

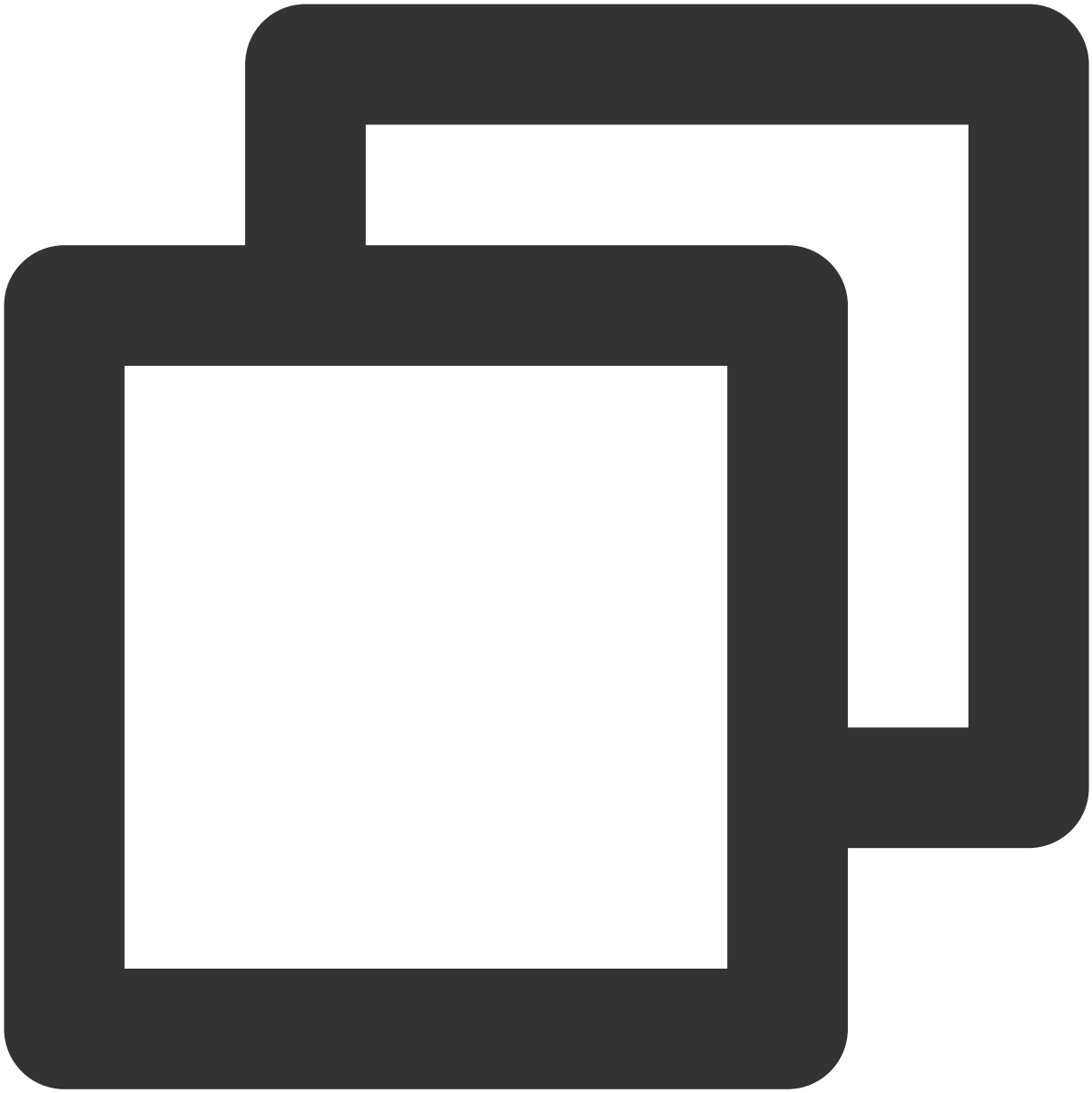
Java 示例

1. 将 Jaeger 库添加到依赖项中。



```
<dependency>
  <groupId>io.jaegertracing</groupId>
  <artifactId>jaeger-client</artifactId>
  <version>0.32.0</version>
</dependency>
```

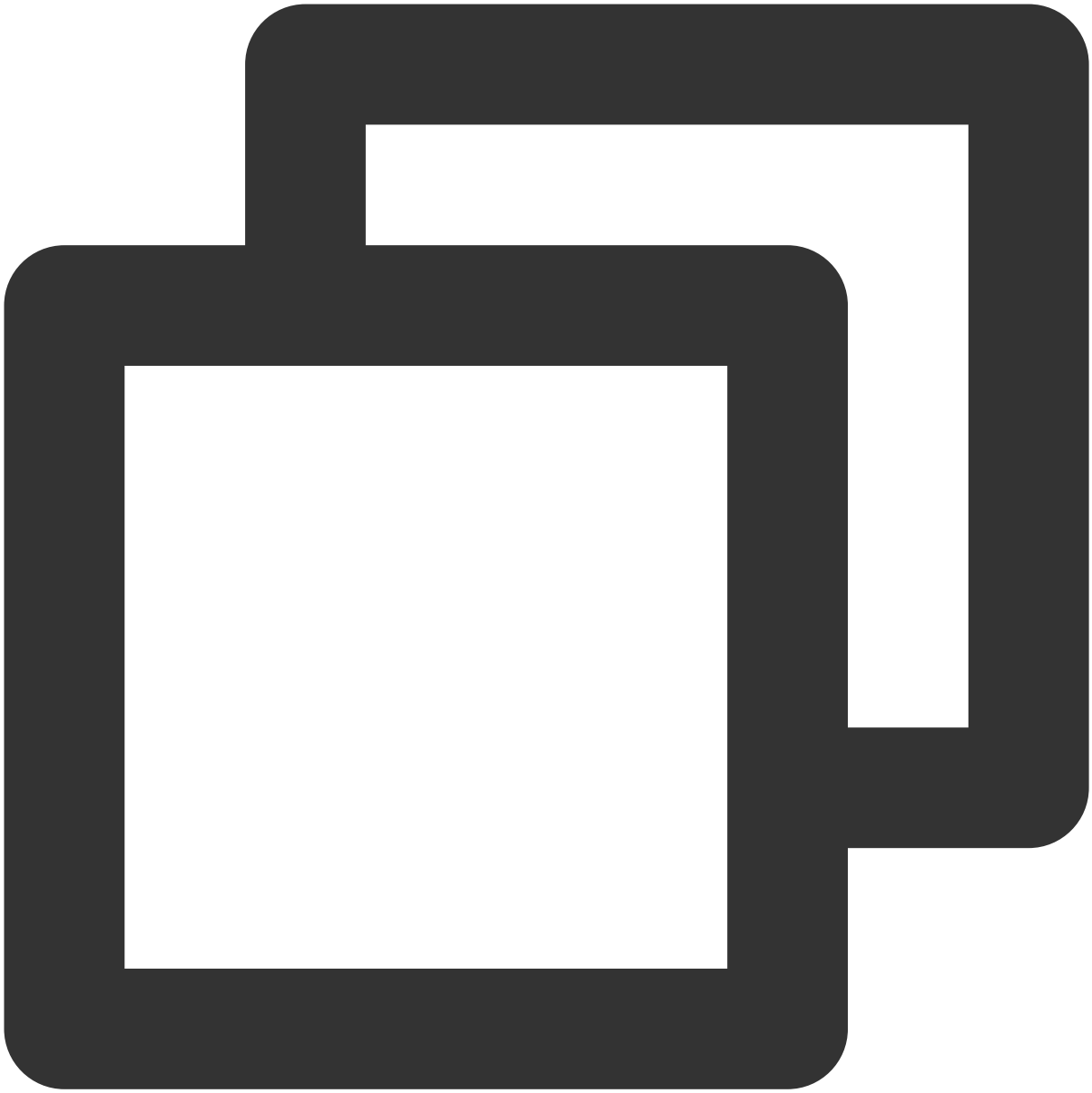
2. 代码示例如下。



```
import io.jaegertracing.Configuration;
import io.jaegertracing.Configuration.ReporterConfiguration;
import io.jaegertracing.Configuration.SamplerConfiguration;
import io.jaegertracing.internal.JaegerTracer;
import io.jaegertracing.internal.samplers.ConstSampler;
import io.opentracing.Span;
import io.opentracing.util.GlobalTracer;

...

SamplerConfiguration samplerConfig = SamplerConfiguration.fromEnv()
```

```
  .withType(ConstSampler.TYPE)
  .withParam(1);

ReporterConfiguration reporterConfig = ReporterConfiguration.fromEnv()
  .withLogSpans(true);

Configuration config = new Configuration("helloWorld")
  .withSampler(samplerConfig)
  .withReporter(reporterConfig);

GlobalTracer.register(config.getTracer());

...
Span parent = GlobalTracer.get().buildSpan("hello").start();
try (Scope scope = GlobalTracer.get().scopeManager()
      .activate(parent)) {
    Span child = GlobalTracer.get().buildSpan("world")
            .asChildOf(parent).start();
    try (Scope scope = GlobalTracer.get().scopeManager()
       .activate(child)) {
    }
}
```

GO 示例



```
import (
    "github.com/opentracing/opentracing-go"
    "github.com/uber/jaeger-client-go"
    "github.com/uber/jaeger-client-go/config"
)

...

func main() {
    ...
    cfg := config.Configuration{
```

```
    Sampler: &config.SamplerConfig{
        Type:  "const",
        Param: 1,
    },
    Reporter: &config.ReporterConfig{
        LogSpans:            true,
        BufferFlushInterval: 1 * time.Second,
    },
    }
    tracer, closer, err := cfg.New(
        "your_service_name",
        config.Logger(jaeger.StdLogger),
    )
    opentracing.SetGlobalTracer(tracer)
    defer closer.Close()

    someFunction()
    ...
}

...

func someFunction() {
    parent := opentracing.GlobalTracer().StartSpan("hello")
    defer parent.Finish()
    child := opentracing.GlobalTracer().StartSpan(
    "world", opentracing.ChildOf(parent.Context()))
    defer child.Finish()
}
```

说明：

更多语言采样说明请参见 Client Library Features。

使用 Skywalking 进行 Java 应用采集配置

最近更新时间：2024-04-02 10:09:04

本文将为您介绍如何使用 Skywalking 进行Java 应用采集配置。

操作背景

在访问量较大时，全链路数据上报可能会导致使用应用性能监控的成本较高。在访问量级较大的情况下，往往会进行数据采样。

说明：

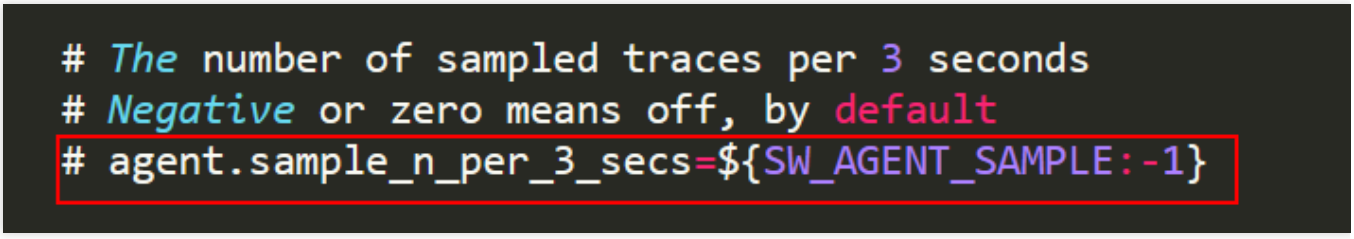
采样指从全量采集的所有链路数据中，采集部分数据进行分析，减少上报数量和链路存储费用，降低使用应用性能监控的成本。

操作前提

已 通过 Skywalking 协议上报 Java 应用。

操作步骤

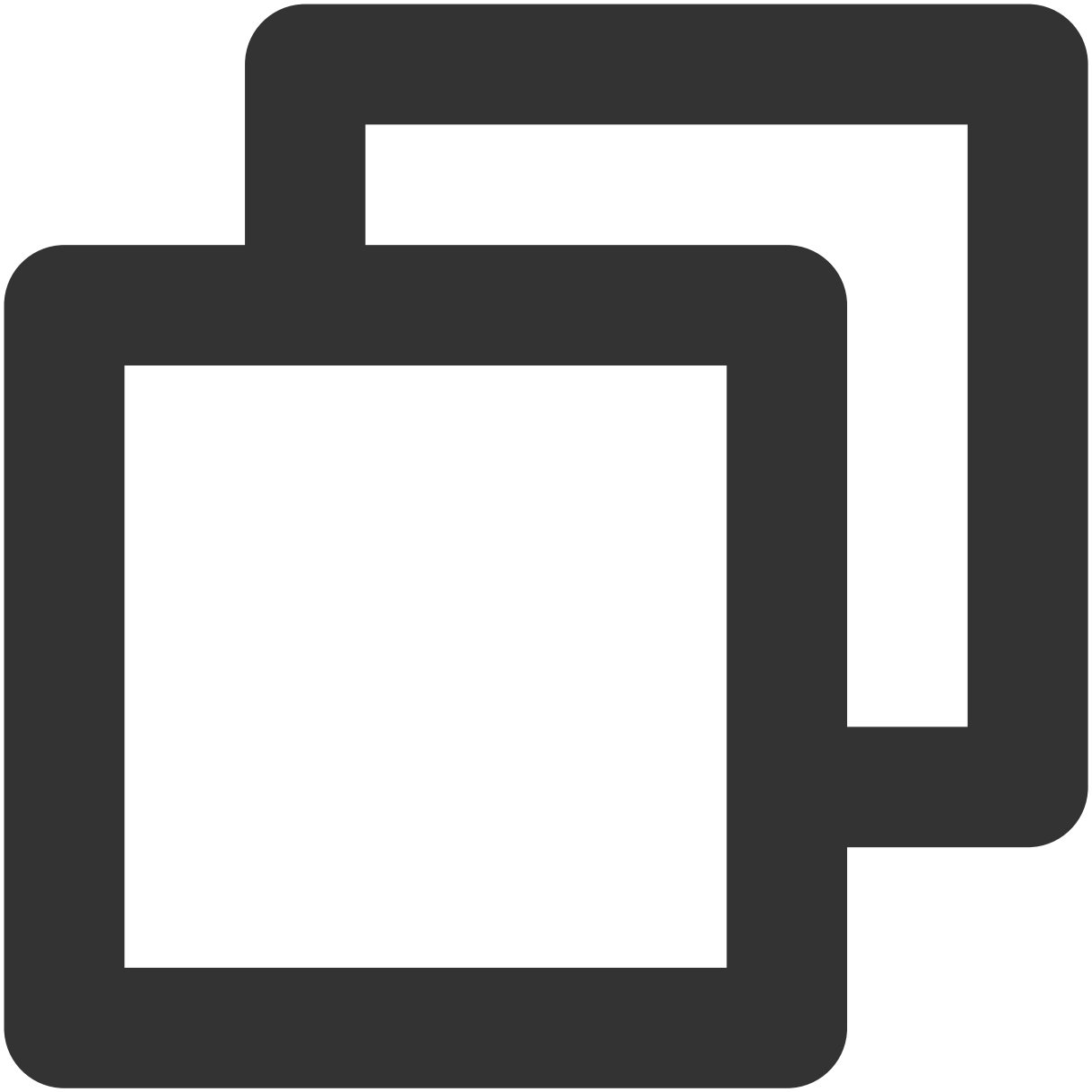
1. 打开 agent/config/agent.config 文件，找到 `agent.sample_n_per_3_secs=${SW_AGENT_SAMPLE:-1}` 配置项。



2. 修改采样率。`agent.sample_n_per_3_secs` 表示设置每 3 秒可收集的链路数据（TraceSegment）的数量。负或零表示全部采样，默认全部采样。

示例：

假设您需要在3秒内采集1500个 TraceSegment ， 可以设置为：



```
agent.sample_n_per_3_secs=${SW_AGENT_SAMPLE:1500}
```