Distributed Tracing

wavelabs

Distributed Tracing

Distributed tracing is a method used to profile and monitor applications built using a microservices architecture. Distributed tracing helps pinpoint where failures occur and what causes poor performance.

Monitoring and Debugging:

- In microservice architecture pattern requests often span multiple services.
- Each service handles a request by performing one or more operations, e.g. database queries, publishes messages, etc.
- External monitoring only tells you the overall response time and number of invocations.
- Log entries for a request are scattered across numerous logs.

Solution

- Assigns each external request a unique external request id.
- Passes the external request id to all services that are involved in handling the request Includes the external request id in all log messages.
- Records information (e.g. start time, end time) about the requests and operations performed when handling a external request in a centralized service.

Zipkin: Spring boot starter for tracing

- Add Zipkin spring boot starter maven dependency.

```
<dependency>
     <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-zipkin</artifactId>
 </dependency>
```

- Configure zipkin url in application properties for posting traces

```
spring.zipkin.base-url=http://localhost:9411/
spring.sleuth.sampler.probability=1
```

- Zipkin use spring.application.name as service name while recording trace information.

This name is used to filter tracing records in Zipkin UI.

Zipkin: Starting Zipkin server

Zipkin server can be started in two ways:

1. Docker: Following docker command starts Zipkin server on port 9411

```
docker run -d -p 9411:9411 openzipkin/zipkin
```

2. Zipkin binary distribution. Following commands downloads Zipkin and start Zipkin on port 9411

```
curl -sSL https://zipkin.io/quickstart.sh | bash -s

java -jar zipkin.jar
```

3. Open the link to view Zipkin dashboard

   http://localhost:9411/zipkin/

Jaeger: Spring boot starter for tracing

- Add Jaeger spring boot starter maven dependency.

```
<dependency>
    <groupId>io.opentracing.contrib</groupId>
    <artifactId>opentracing-spring-jaeger-cloud-starter</artifactId>
    <version>1.0.3</version>
</dependency>
```

- Configure Jaeger configuration in application properties

```
opentracing.jaeger.udp-sender.host=127.0.0.1
opentracing.jaeger.udp-sender.port=6831
opentracing.jaeger.service-name=rest-consumer-service
opentracing.jaeger.const-sampler.decision=true
opentracing.jaeger.probabilistic-sampler.sampling-rate =1.0
opentracing.jaeger.rate-limiting-sampler.max-traces-per-second = 1.0
```

Jaeger: Starting Jaeger server

- The following docker command will start Jaeger server:

```
docker run -d --name jaeger \
  -e COLLECTOR_ZIPKIN_HTTP_PORT=9411 \
  -p 5775:5775/udp \
  -p 6831:6831/udp \
  -p 6832:6832/udp \
  -p 5778:5778 \
  -p 16686:16686 \
  -p 14268:14268 \
  -p 9411:9411 \
  jaegertracing/all-in-one:1.8
```

- To view Jaeger dashboard http://localhost:16686/search