

聊聊skywalking的TraceSegmentServiceClient

 **codecraft** 发布于 2020-03-02

序

本文主要研究一下skywalking的TraceSegmentServiceClient

TracingContextListener

skywalking-6.6.0/apm-sniffer/apm-agent-core/src/main/java/org/apache/skywalking/apm/agent/core/context/TracingContextListener.java

```
public interface TracingContextListener {
    void afterFinished(TraceSegment traceSegment);
}
```

- TracingContextListener定义了afterFinished方法，其参数为TraceSegment

TraceSegment

skywalking-6.6.0/apm-sniffer/apm-agent-core/src/main/java/org/apache/skywalking/apm/agent/core/context/trace/TraceSegment.java

```
public class TraceSegment {

    private ID traceSegmentId;

    private List<TraceSegmentRef> refs;

    private List<AbstractTracingSpan> spans;

    private DistributedTraceIds relatedGlobalTraces;

    private boolean ignore = false;

    private boolean isSizeLimited = false;

    private final long createTime;

    public TraceSegment() {
        this.traceSegmentId = GlobalIdGenerator.generate();
        this.spans = new LinkedList<AbstractTracingSpan>();
        this.relatedGlobalTraces = new DistributedTraceIds();
        this.relatedGlobalTraces.append(new NewDistributedTraceId());
        this.createTime = System.currentTimeMillis();
    }

    public void ref(TraceSegmentRef refSegment) {
        if (refs == null) {
```

- TraceSegment定义了traceSegmentId、refs、spans、relatedGlobalTraces等属性；它提供了ref、relatedGlobalTraces、archive、finish、transform等方法

IConsumer

skywalking-6.6.0/apm-commons/apm-datacarrier/src/main/java/org/apache/skywalking/apm/commons/datacarrier/consumer/IConsumer.java

```
public interface IConsumer<T> {
    void init();

    void consume(List<T> data);

    void onError(List<T> data, Throwable t);

    void onExit();
}
```

- IConsumer定义了init、consume、onError、onExit方法

TraceSegmentServiceClient

skywalking-6.6.0/apm-sniffer/apm-agent-core/src/main/java/org/apache/skywalking/apm/agent/core/remote/TraceSegmentServiceClient.java

```
@DefaultImplementor
public class TraceSegmentServiceClient implements BootService, IConsumer<TraceSegment>, TracingContextListener,
GRPCChannelListener {
    private static final ILog logger = LogManager.getLogger(TraceSegmentServiceClient.class);
    private static final int TIMEOUT = 30 * 1000;

    private long lastLogTime;
    private long segmentUplinkedCounter;
    private long segmentAbandonedCounter;
    private volatile DataCarrier<TraceSegment> carrier;
    private volatile TraceSegmentReportServiceGrpc.TraceSegmentReportServiceStub serviceStub;
    private volatile GRPCChannelStatus status = GRPCChannelStatus.DISCONNECT;

    @Override
    public void prepare() throws Throwable {
        ServiceManager.INSTANCE.findService(GRPCChannelManager.class).addChannelListener(this);
    }

    @Override
    public void boot() throws Throwable {
        lastLogTime = System.currentTimeMillis();
        segmentUplinkedCounter = 0;
        segmentAbandonedCounter = 0;
        carrier = new DataCarrier<TraceSegment>(CHANNEL_SIZE, BUFFER_SIZE);
        carrier.setBufferStrategy(BufferStrategy.IF_POSSIBLE);
        carrier.consume(this, 1);
    }
}
```

- TraceSegmentServiceClient实现了BootService、IConsumer、TracingContextListener、GRPCChannelListener接口；其prepare方法往GRPCChannelManager注册自身的channelListener；其boot方法设置lastLogTime，实例化DataCarrier，并设置其consumer为自身；其onComplete方法执行TracingContext.ListenerManager.add(this)；其shutdown方法执行TracingContext.ListenerManager.remove(this)以及carrier.shutdownConsumers()；其consume方法在status为CONNECTED的时候执行upstreamSegmentStreamObserver.onNext(upstreamSegment)、upstreamSegmentStreamObserver.onCompleted()以及status.wait4Finish()；其afterFinished方法执行carrier.produce(traceSegment)；其statusChanged设置serviceStub及status

ConsumerThread

skywalking-6.6.0/apm-commons/apm-datacarrier/src/main/java/org/apache/skywalking/apm/commons/datacarrier/consumer/ConsumerThread.java

```
public class ConsumerThread<T> extends Thread {
    private volatile boolean running;
    private IConsumer<T> consumer;
    private List<DataSource> dataSources;
    private long consumeCycle;

    ConsumerThread(String threadName, IConsumer<T> consumer, long consumeCycle) {
        super(threadName);
        this.consumer = consumer;
        running = false;
        dataSources = new ArrayList<DataSource>(1);
        this.consumeCycle = consumeCycle;
    }

    /**
     * add whole buffer to consume
     *
     * @param sourceBuffer
     */
    void addDataSource(QueueBuffer<T> sourceBuffer) {
        this.dataSources.add(new DataSource(sourceBuffer));
    }

    @Override
    public void run() {
        running = true;
```

- ConsumerThread继承了Thread，其run方法会循环执行consume(consumeList)，跳出循环时会再次执行consume(consumeList)，最后执行consumer.onExit()；consume方法会遍历dataSources，执行其dataSource.obtain(consumeList)，然后在consumeList不为空的时候执行consumer.consume(consumeList)方法

ConsumeDriver

skywalking-6.6.0/apm-commons/apm-

datacarrier/src/main/java/org/apache/skywalking/apm/commons/datacarrier/consumer/ConsumeDriver.java

```
public class ConsumeDriver<T> implements IDriver {
    private boolean running;
    private ConsumerThread[] consumerThreads;
    private Channels<T> channels;
    private ReentrantLock lock;

    public ConsumeDriver(String name, Channels<T> channels, Class<? extends IConsumer<T>> consumerClass, int num,
        long consumeCycle) {
        this(channels, num);
        for (int i = 0; i < num; i++) {
            consumerThreads[i] = new ConsumerThread("DataCarrier." + name + ".Consumser." + i + ".Thread",
                getNewConsumerInstance(consumerClass), consumeCycle);
            consumerThreads[i].setDaemon(true);
        }
    }

    public ConsumeDriver(String name, Channels<T> channels, IConsumer<T> prototype, int num, long consumeCycle) {
        this(channels, num);
        prototype.init();
        for (int i = 0; i < num; i++) {
            consumerThreads[i] = new ConsumerThread("DataCarrier." + name + ".Consumser." + i + ".Thread",
                prototype, consumeCycle);
            consumerThreads[i].setDaemon(true);
        }
    }
}
```

- ConsumeDriver实现了IDriver接口，其ConsumeDriver会创建num个ConsumerThread；其begin方法会执行allocateBuffer2Thread，给每个consumerThread添加dataSource，然后执行consumerThread.start()；其close方法会执行consumerThread.shutdown()

小结

TraceSegmentServiceClient实现了BootService、IConsumer、TracingContextListener、GRPCChannelListener接口；其prepare方法往GRPCChannelManager注册自身的channelListener；其boot方法设置lastLogTime，实例化DataCarrier，并设置其consumer为自身；其onComplete方法执行TracingContext.ListenerManager.add(this)；其shutdown方法执行TracingContext.ListenerManager.remove(this)以及carrier.shutdownConsumers()；其consume方法在status为CONNECTED的时候执行upstreamSegmentStreamObserver.onNext(upstreamSegment)、upstreamSegmentStreamObserver.onCompleted()以及status.wait4Finish()；其afterFinished方法执行carrier.produce(traceSegment)；其statusChanged设置serviceStub及status

doc

- [TraceSegmentServiceClient](#)

[skywalking](#)

阅读 565 • 更新于 2020-03-02



本作品系原创，采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



code-craft

spring boot , docker and so on 欢迎关注微信公众号: geek_luandun

关注专栏



codecraft

当一个代码的工匠回首往事时，不因虚度年华而悔恨，也不因碌碌无为而羞愧，这样，当他老的时候，可以很自豪告诉世人，我曾经将代码注入生命去打造互联网的浪潮之巅，那是个很疯狂的时代，我在一波波的浪潮上留下了或重如泰山或轻如鸿毛的几笔。

11.5k 声望 1.9k 粉丝

关注作者

0 条评论

得票数

最新



撰写评论 ...



提交评论

你知道吗？

修正程序错误的第一步是要复现这个错误。

注册登录

继续阅读

聊聊skywalking的jdk-threading-plugin

[skywalking-6.6.0/apm-sniffer/bootstrap-plugins/jdk-threading-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 567](#)

聊聊skywalking的kafka-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/kafka-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 991](#)

聊聊skywalking的http-async-client-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/httpasyncclient-4.x-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 563](#)

聊聊skywalking的jedis-pulgin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/jedis-2.x-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 813](#)

聊聊skywalking的mysql-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/mysql-8.x-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 607](#)

聊聊skywalking的lettuce-plugin

[skywalking-6.6.0/apm-sniffer/optional-plugins/lettuce-5.x-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 597](#)

聊聊skywalking的JVMService

[skywalking-6.6.0/apm-sniffer/apm-agent-core/src/main/java/org/apache/skywalking/apm/agent/core/boot/BootService.java](#)

[codecraft](#) · [阅读 616](#)

聊聊skywalking的sharding-sphere-plugin

[skywalking-6.6.0/apm-sniffer/apm-sdk-plugin/sharding-sphere-4.x-plugin/src/main/resources/skywalking-plugin.def](#)

[codecraft](#) · [阅读 750](#)

产品

[热门问答](#)

[热门专栏](#)

[热门课程](#)

[最新活动](#)

[技术圈](#)

[酷工作](#)

课程

[Java 开发课程](#)

[PHP 开发课程](#)

[Python 开发课程](#)

[前端开发课程](#)

[移动开发课程](#)

资源

[每周精选](#)

[用户排行榜](#)

[勋章](#)

[帮助中心](#)

[声望与权限](#)

[社区服务中心](#)

[建议反馈](#)

合作

[关于我们](#)

[广告投放](#)

[职位发布](#)

[讲师招募](#)

[联系我们](#)

[合作伙伴](#)

关注

[产品技术日志](#)

[社区运营日志](#)

[市场运营日志](#)

[团队日志](#)

[社区访谈](#)

条款

[服务协议](#)

[隐私政策](#)

[下载 App](#)