

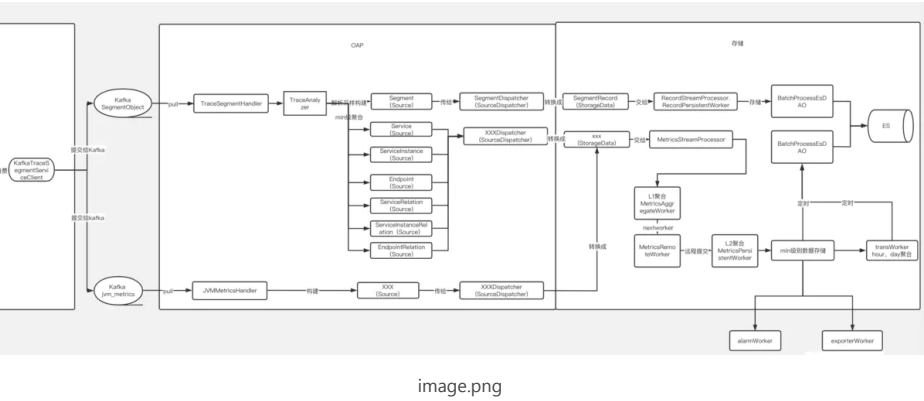
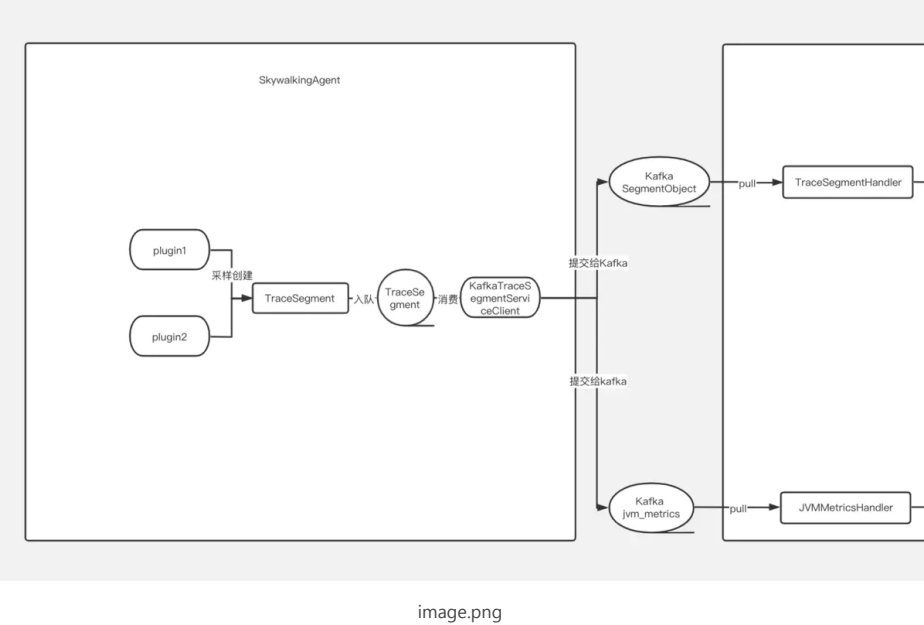
14、Skywalking的OAP-核心流程串烧

 rock_fish 关注

2021.06.09 15:38:01 字数 710 阅读 124

1. Kafka/Grpc的 Handler接收数据,解析接收到的数据构建成 Source,不同的数据类型,这个过程复杂程度不同。
2. DispatcherManager 找到 Source 对应的 SourceDispatcher,将 Source 传入 dispatch 方法中做处理
- 2.1 将 Source 转换成对应的 StorageData 的子类,比如 Segment (Source) 转换成 SegmentRecord (StorageData)
- 2.2 调用具体的 StreamProcessor 处理 StorageData

流程图: <https://www.processon.com/view/link/60c1e5f20791297a3f0e4411>




Kafka接收数据

大致的流程为: Handler->解析构建Source-> SourceDispatcher#dispatch

有哪些SourceDisptch呢,他们都处理什么类型的数据呢,这里是通过扫描的方式来实现的,其



 rock_fish 关注

总资产6

21、Skywalking的埋点-Agent动态采样控制
阅读 219

22、skywalking的Trace数据协议
阅读 12

skywalking的日常维护1:
com.netflix.zuul.exception.ZuulExcept
阅读 41

推荐阅读

[Spring MVC]HandlerMapping的初始化
阅读 266

Mybatis实现原理
阅读 505

数据绑定
阅读 205

Java2021高频面试题(含答案, 适合面试前冲刺一下快速记忆)
阅读 3,530

MyBatis的架构原理
阅读 1,082



比如 `SourceDispatcher#dispatch` 处理的 `Source` 的类型是 `Segment`：

```
1 public class SegmentDispatcher implements SourceDispatcher<Segment> {
2
3     @Override
4     public void dispatch(Segment source) {
5         ...
6     }
7 }
```

这些被扫描到的实例，会被存储到 `DispatcherManager` 的成员变量 `dispatcherMap` 中，其key是 `Source#scope()`，value是具体的 `SourceDispatcher` 实现类的实例。在使用的时候通过 `Source#scope()` 在 `dispatcherMap` 中检索到Source对应的xxxDispatcher实例，进而调用其dispatch方法

需要知晓 `SourceDispatcher` 的部分实现是通过 OAL 脚本生成的,我们在源码中看不到

```
1 BrowserAppTrafficSourceDispatcher.dispatch(SOURCE)
2 BrowserErrorLogRecordDispatcher.dispatch(BrowserErrorLog)
3 DatabaseStatementDispatcher.dispatch(DatabaseSlowStatement)
4 EndpointCallRelationDispatcher.dispatch(EndpointRelation)
5 EndpointMetaDispatcher.dispatch(EndpointMeta)
6 EndpointTrafficDispatcher.dispatch(Endpoint)
7 InstanceTrafficDispatcher.dispatch(ServiceInstance)
8 InstanceUpdateDispatcher.dispatch(ServiceInstanceUpdate)
9 LogRecordDispatcher.dispatch(Log)
10 NetworkAddressAliasSetupDispatcher.dispatch(NetworkAddressAliasSetup)
11 SegmentDispatcher.dispatch(Segment)
12 ServiceCallRelationDispatcher.dispatch(ServiceRelation)
13 ServiceInstanceCallRelationDispatcher.dispatch(ServiceInstanceRelation)
14 ServiceMetaDispatcher.dispatch(ServiceMeta)
15 ServiceTrafficDispatcher.dispatch(Service)
16 ZipkinSpanRecordDispatcher.dispatch(ZipkinSpan)
```

存储

启动时扫描@Stream标注，这个注解标注在StorageData上，看其核心成员：

- name：这个StorageData在es中对应的索引的名称，或者别名
- processor：构建数据存储处理的链路，接收数据，调用worker，进而执行存储。
- builder：写时：StorageData->map->IndexRequest / 读时：IndexResponse->map->StorageData

例如：

```
1 @Stream(name = LogRecord.INDEX_NAME, scopeId = DefaultScopeDefine.LOG, builder = LogRecord.Bui
2 public class LogRecord extends AbstractLogRecord {
```

扫描@Stream，对其解析后会构造StreamProcessor，调用其create方法 做一些数据存储层的初始化工作：

- 创建model，model 的关键作用：创建索引和索引模板
- 创建dao，dao是数据访问层
- 创建 StorageBuilder，StorageBuilder是数据转换，存储中间件的对象与上层StorageData之间在读写操作时的互相转换。
- 创建worker链，数据处理逻辑，这里会调用dao，执行存储。

这些内容的工作链路大致如下：

1. 启动时扫描@Stream标注，如果不需要创建索引和模板，就跳过；如果需要创建索引和模板，就执行创建索引和模板的操作。

写下你的评论...

评论0

赞



- 2、遍历workder链，寻找对应StorageData的worker ,调用其in方法继续处理
- 3、通常在in方法中调用dao执行存储
- 4、dao中调用StorageBuilder的转换方法，将StorageData转换成map，进而拼装成es 的IndexRequest对象。

0人点赞 >

监控

更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下

rock_fish

总资产6 共写了9.4W字 获得82个赞 共27个粉丝

关注

现在停车棚一般都用膜结构的，你知道吗？



推荐阅读

更多精彩内容 >

转载springboot核心属性配置

原文地址<https://www.cnblogs.com/dsn727455218/p/9304490.html> #...

zz云飞扬

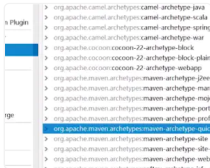
阅读 175 评论 0 赞 0

电影售票系统开发流程及其bug修复日志--业务开发(1)

传统业务 传统业务应用基本把所有的东西都集合在一起，传统应用带来的问题，单一业务的开发和迭代困难，这个时候牵扯到两...

冒绿光的盒子

阅读 198 评论 0 赞 0




艺术签名在线生成

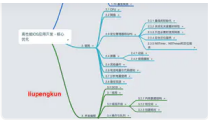


写下你的评论...

评论0 赞


此篇博客是《高性能iOS应用开发》一书第二部分“核心优化”的读书笔记，主要包括“内存管理”、“能耗(电量消耗)”、...

 Q以梦为马 阅读 736 评论 0 赞 14



Spring Boot application.properties

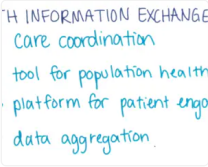
原文链接：<https://docs.spring.io/spring-boot/docs/1.4.x/refere...>

 pseudo_niaonao 阅读 3,802 评论 0 赞 9

Intro to Health Informatics 第三周笔记

Lesson 3: Health Information Exchange Reasons for establi...

 我的名字叫清阳 阅读 9,719 评论 0 赞 12



写下你的评论...

评论0

赞