

履约后链路压测在SLS实施

0 项目计划

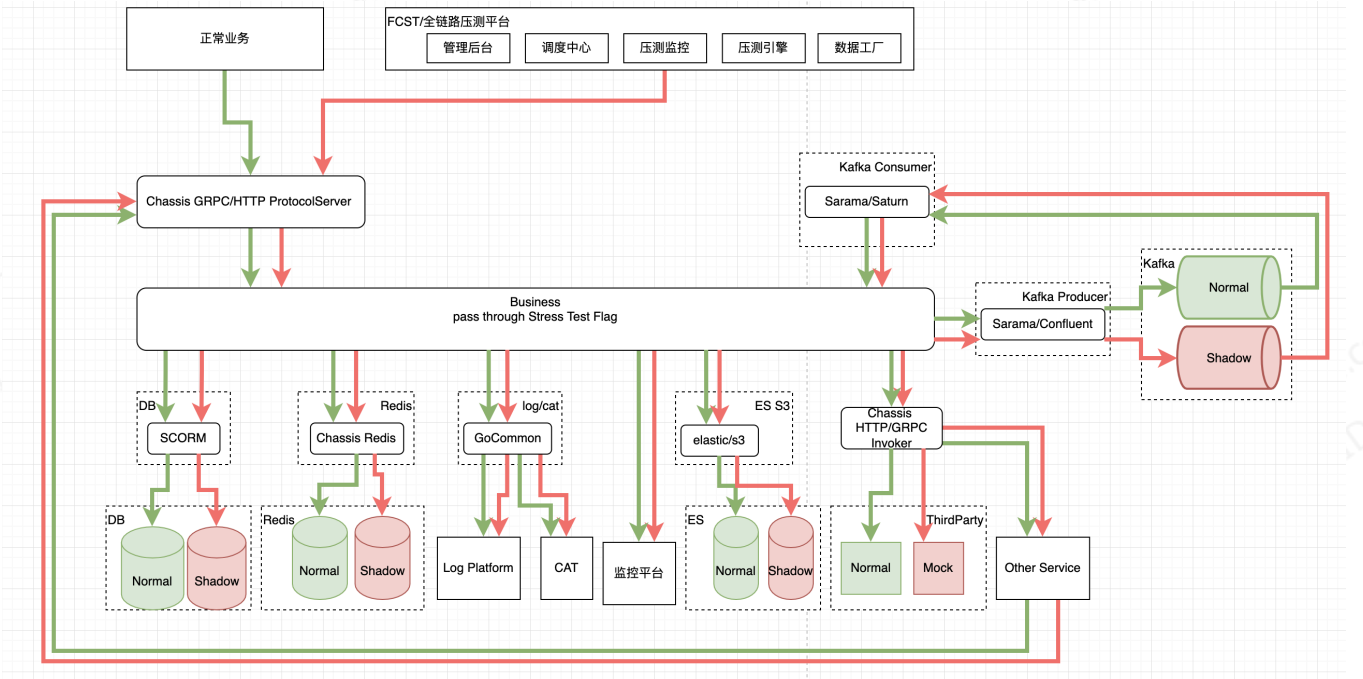
	子任务	jira	PIC
<div> SPOMS-18527 - jira 问题不存在或者您没有权限查看。</div> <div>项目明细</div>	压测平台写流量模式改造	lin.zhu@shopee.com	
	SLS下单链路影子库	lin.zhu@shopee.com	
	SLS轨迹链路影子库	lin.zhu@shopee.com	
	组件改造-LPS	wei.zhongzw@shopee.com	
	组件改造-LFS	lening.zeng@shopee.com	
	组件改造-LLS	lening.zeng@shopee.com	
	组件改造-PIS	chay.cao@shopee.com	
	组件改造-LTS	jingdong.xu@shopee.com	
	组件改造-FINANCE	guifeng chen Qiuru Chen	
	组件改造-WAYBILLCENTER	yanling.wu@shopee.com	
	cat链路检查工具	lin.zhu@shopee.com	
	项目组件检查工具	lin.zhu@shopee.com	

1 摘要

1.1 项目目的

目前SLS压测方式仅能覆盖履约前链路，缺乏手段排查履约后链路系统隐患和容量。本项目对SLS项目进行改造，以提供一套将流量与业务数据区分的压测设施。

全链路压测架构图（参考 1. 全链路压测业务技术改造指南）：



1.2 参考资料

[全链路压测](#)

[1. 全链路压测业务技术改造指南](#)

[002.Shopee MySQL Shadow Databases](#)

[B2C发货场景业务梳理](#)

1.3 术语与缩略词

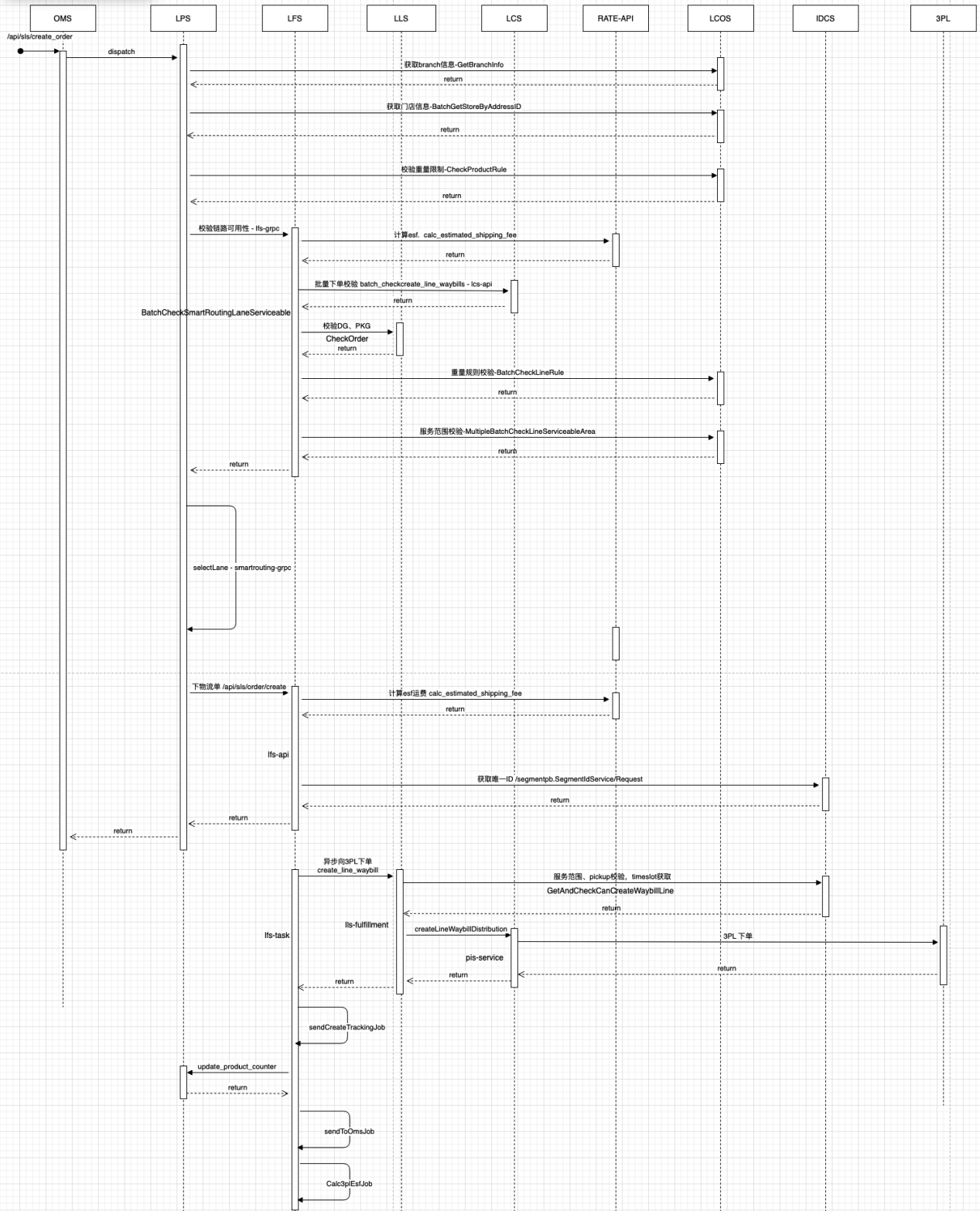
名次	释义	备注
影子请求	走入压测链路的请求，对线上实际环境不产生任何影响。	
影子流量	压测链路的流量，与实际流量完全隔离开。	

2 业务分析

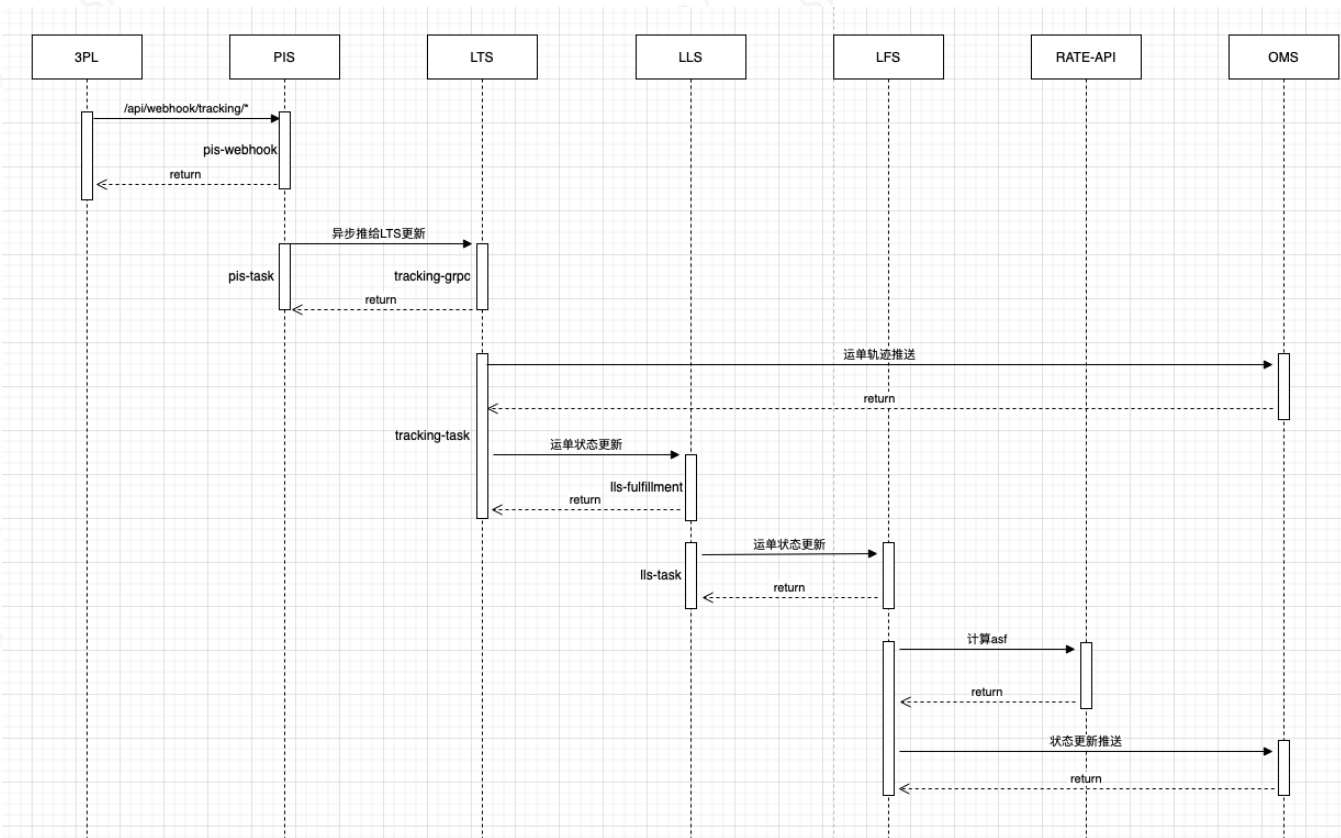
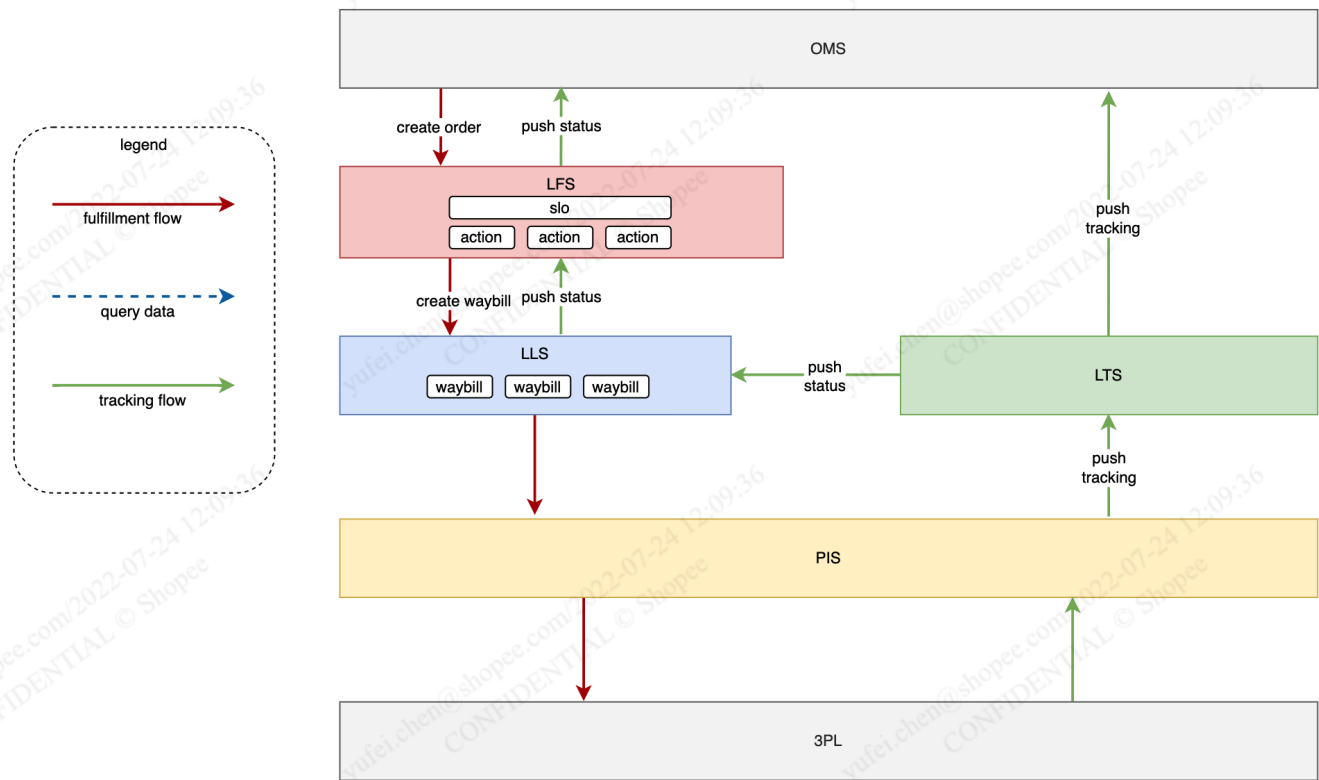
履约后主要分为 create order和tracking两条链路，我们分别分析这两个链路的业务流程、涉及服务。

2.1 create order链路

iShot (购买解锁去水印)



2.2 tracking update链路



发货序列图

3 系统分析

3.1 压测流量

@todo 如何制作多场景、真实的压测流量

参考: [全链路压测#数据构造](#)

参考: [数据工厂](#)

3.2 压测数据隔离：影子库

为了模拟真实情况，一般选取和live相同的环境进行压测，这些发送到live环境的压测流量称为**影子流量**。

为了避免压测影响线上业务，系统服务需要识别出影子流量，将影子流量路由到**影子库**，或做mock、drop处理。

参考: [流量识别及传输功能设计说明](#)

参考: [全链路压测-压测数据隔离：影子库](#)

3.3 流量污染

引入影子流量后，在影子标识的传递过程中存在以下逻辑：

- 1.只有在context内有影子标志位且值为true的情况下，才会被识别为影子流量。除此之外，都会被当作正常流量来处理。
- 2.如果context内没有影子标志位，比如使用的是context.background(), 则中间件会上报一个unsafeContext 的event到cat，但仍然会当作正常流量进行链路履约。

下面对问题进行分析。

问题	影响	应对方式
影子流量的ctx在链路中断，比如使用了context.background(), 写入到了正常链路，被当作正常流量履约	1.影子流量写入正常表，或执行SQL报错率变高。 2.影子流量写入正常topic，导致队列阻塞。 3.发起外部请求时，没有被mock。	1.上线前进行验证，保证压测涉及到的链路是不会产生ctx中断的。 2.产生此问题时，原始的ctx已经丢失，不能判断是否影子流量， 需要对 order Id进行改造加上特殊标识 ，如果线上压测时产生了此问题，可以快速定位到数据。 3. 对于unsafeContext中断处理，在go live前发现隐患。
正常流量的ctx在链路中断，比如使用了context.background(), 写入到了正常链路，被当作正常流量履约	无影响	
影子流量的ctx标志位由true变为false	不会出现，chassis没有暴露修改影子标记的方法。	-
正常流量的ctx标志位由false变为true	不会出现，chassis没有暴露修改影子标记的方法。	

3.4 压测安全

不仅流量污染，压测时还经常会因为性能容量问题导致服务过载，继而引发生产故障。

压测安全是整个压测过程中非常重要关键的一环，我们需要在可能出现问题前采取措施，避免影响扩大。

为此我们需要建设压测安全机制，主要包括以下内容：

- 1. 压测前检查，所有检查项通过压测任务才可以启动

2. 压测时熔断，中止发送压测流量。参考：[全链路压测平台--技术文档-压测熔断](#)
3. 服务自我保护，丢弃压测流量

4 任务事项

经过上述业务分析和系统分析，我们得到履约后链路压测的改造事项。

我们将履约后链路分为create_order和tracking两部分，分别进行描述。首先给出其中的基础：组件改造overview

4.0 改造组件overview

chassis升级到v0.4.0
使用chassis http客户端
使用chassis scorm客户端，并配置影子库
使用chassis redis客户端，无需配置影子库
使用chassis saturn客户端，配置影子topic和消费任务
layercache改造，支持影子标记
使用chassis log组件
TIDB mock
3PL mock

参考：[1. 全链路压测业务技术改造指南](#)

4.1 create order链路改造

我们挑选ID市场80020渠道做为第一期改造参考用例，准备压测数据

压测数据准备

市场	shipping_channel_id	3PL	参数模版	PIC
ID	80029	SPX	create order参数模版	<input type="checkbox"/> lin.zhu@shopee.com

使用特殊前缀随机生成ordersn和forderid，替换参数模版中对应的字段，逻辑如下：

```
baseInfo["ordersn"] = "FCST" + strconv.Itoa(int(randInstance.Int63n(999999999999)))
baseInfo["forderid"] = "666666" + strconv.Itoa(int(randInstance.Int63n(999999999999)))
```

涉及服务

服务名	PIC	
lps-livetestapi-live	wei.zhongzw@shopee.com	<input type="checkbox"/> 组件改造
lfs-livetestapi-live	lening.zeng@shopee.com	<input type="checkbox"/> 组件改造
lfs-tasklivetest-live	lening.zeng@shopee.com	<input type="checkbox"/> 组件改造

lfs-fulfillmentlivetest-live	lening.zeng@shopee.com	<input type="checkbox"/> 组件改造
pis-pisservicelivetest-live	chay.cao@shopee.com	<input type="checkbox"/> 组件改造
waybillcenter-staticlivetest-live	Yanling.liu	<input type="checkbox"/> 组件改造

db

服务名	db name	PIC	
lfs-livetestapi-live	shadow_lfs_order_id_db	lening.zeng@shopee.com	<input type="checkbox"/> 影子db配置
lfs-tasklivetest-live	shadow_ssc_lfs_order_id_db	lening.zeng@shopee.com	<input type="checkbox"/> 影子db配置
lfs-fulfillmentlivetest-live	shadow_lcs_v2_id_db	lening.zeng@shopee.com	<input type="checkbox"/> 影子db配置
waybillcenter-staticlivetest	shadow_ssc_static_data_id_db	Yanling.liu	<input type="checkbox"/> 影子db配置

codis

使用live codis，组件在key加上shadow前缀

kafka

服务名	topic	PIC	
lfs-livetestapi-live	shadow_lfs.live.id.order.create	lening.zeng@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务
lfs-tasklivetest-live	shadow_lfs.live.id.order.create	lening.zeng@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务

4.2 tracking update链路

数据

挑选lcs_channel_id = 8037 的18组轨迹参数做为模版

市场	shipping_channel_id	3PL	lcs_channel_id	参数模版	PIC
ID	80029	SPX	8037	tracking参数模版	<input type="checkbox"/> lin.zhu@shopee.com

在下单时，lfs-fulfillment把3PLTN mock成OrderSN存储入库

压测平台使用下单时的OrderSN替换tracking参数模版中的carrier_tn得到tracking参数保存下来，供tracking压测使用

涉及服务

服务名	PIC	
pis-piswebhooklivetest-live	chay.cao@shopee.com	<input type="checkbox"/> 组件改造
sls-pistasklivetest-live	chay.cao@shopee.com	<input type="checkbox"/> 组件改造
sls-trackinggrpclivetest-live	jingdong.xu@shopee.com	<input type="checkbox"/> 组件改造
sls-trackingtasklivetest-live	jingdong.xu@shopee.com	<input type="checkbox"/> 组件改造
lts-tasklivetest-live	lening.zeng@shopee.com	<input type="checkbox"/> 组件改造
lts-querylivetest-live	lening.zeng@shopee.com	<input type="checkbox"/> 组件改造

db

服务名	db name	PIC	
pis-piswebhooklivetest-live	shadow_pis_tracking_id_db	chay.cao@shopee.com	<input type="checkbox"/> 影子db配置
sls-pistasklivetest-live	shadow_partner_integration_db	chay.cao@shopee.com	<input type="checkbox"/> 影子db配置
sls-trackinggrpclivetest-live	shadow_ssc_lts_tracking_id_db	jingdong.xu@shopee.com	<input type="checkbox"/> 影子db配置
sls-trackingtasklivetest-live	shadow_ssc_lts_id_db	jingdong.xu@shopee.com	<input type="checkbox"/> 影子db配置
lts-tasklivetest	shadow_lcs_id_db shadow_lcs_site_id_db shadow_lcs_site_v2_id_db	lening.zeng@shopee.com	
lts-querylivetest		lening.zeng@shopee.com	

codis

使用live codis，组件在key加上shadow前缀

kafka

服务名	topic	负责人	
pis-piswebhooklivetest-live	shadow_sls-pis-trackingpull-live-id	chay.cao@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务

sls-pistasklivetest	shadow_sls-pis-trackingpush-live-id	chay.cao@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务
sls-trackinggrpclivetest	shadow_sls-lts-sloupdatetracking-live-id shadow_sls-lts-create-by-event-live-id	jingdong.xu@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务
sls-trackingtasklivetest	shadow_sls-lts-sloupdatetracking-live-id shadow_sls-lts-slo-oms-notify-live-id shadow_sls-lts-slo-lts-notify-live-id shadow_sls-lts-slo-spx-notify-live-id shadow_sls-lts-create-by-event-live-id	jingdong.xu@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务
lts-tasklivetest	shadow_sls-lts-push-tracking-status-live-idx shadow_sls-lts-push-weight-fee-live-id shadow_sls-lts-update-waybill-status-live-id shadow_sls-lts-update-weight-fee-live-id	lening.zeng@shopee.com	<input type="checkbox"/> 配置影子topic和消费任务

4 检查工具

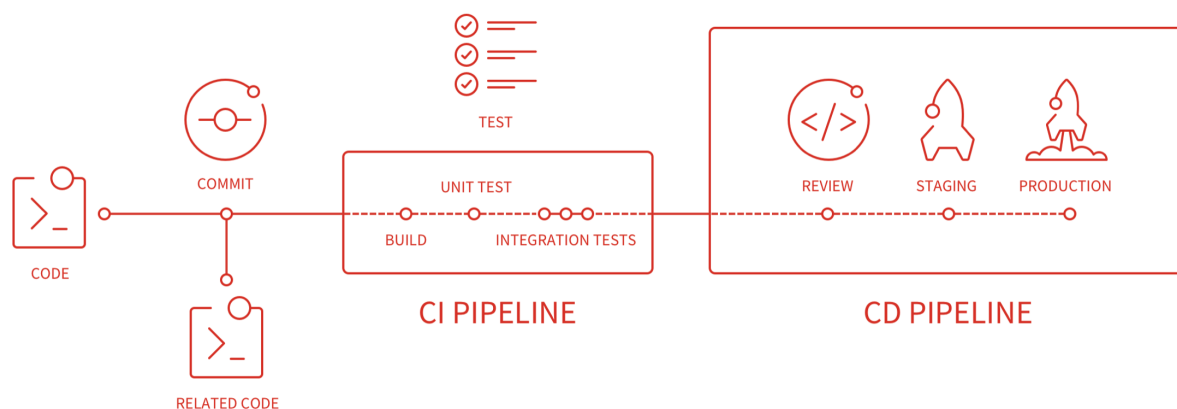
3.1 组件检查

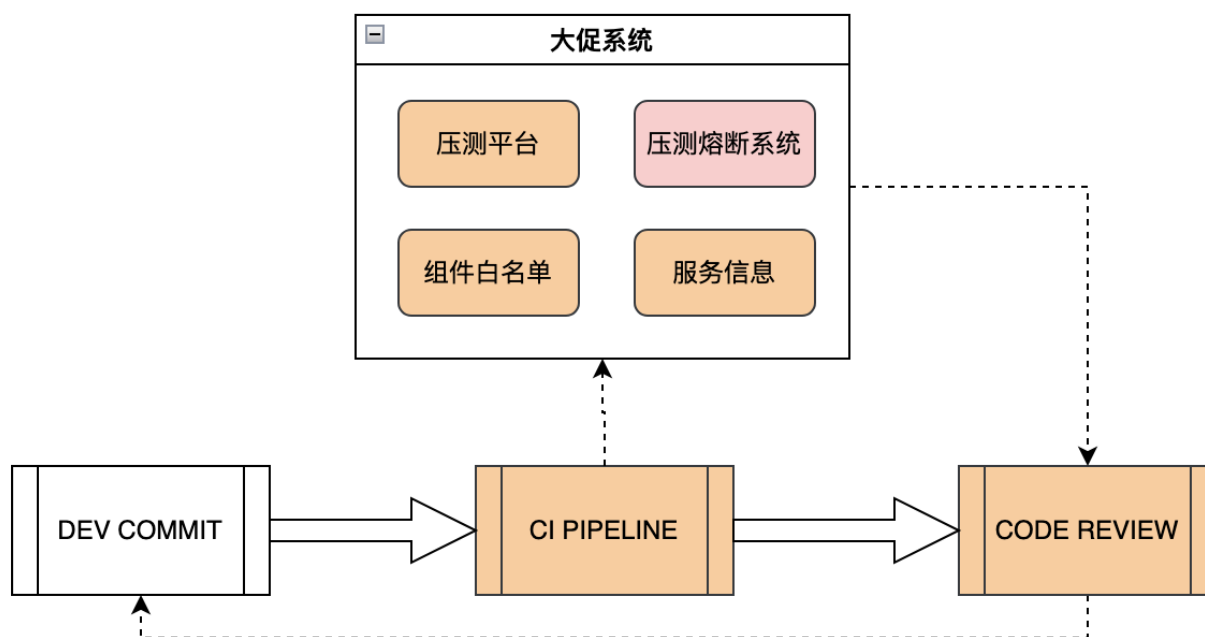
为保障影子 **流量被正确路由、链路上报准确**，我们需要使用标准组件。

随着代码迭代、架构升级，不断可能会有预期外的依赖组件引入。

所以我们需要持续扫描各个服务的组件使用情况，分析组件及其版本是否在白名单内。如发现预期外的组件出现，及早作出提示并熔断压测平台，直到组件及其版本符合预期。

我们选择在GITLAB CI PIPELINE阶段检查各个服务组件情况，并上报到大促系统。大促系统做汇总展示，并熔断压测平台，避免因组件导致压测故障。





- 在gitlab配置ci pipeline, merge时执行命令检查依赖信息

```
go list -m -f '{{if not .Indirect}}{{.Path}} {{.Version}}{{end}}' all
```

可以得到

```
git.garena.com/shopee/bg-logistics/logistics/celestial
git.garena.com/shopee/bg-logistics/go/chassis v0.3.2-a3.0.20220902103959-02c2af1e346e
git.garena.com/shopee/bg-logistics/go/gocommon v0.4.0-b1
git.garena.com/shopee/bg-logistics/go/scorm v0.0.22
git.garena.com/shopee/bg-logistics/service/saturn-rpc-job v1.1.0
github.com/aws/aws-sdk-go v1.38.61
github.com/bsm/redislock v0.7.0
github.com/confluentinc/confluent-kafka-go v1.4.2
github.com/dgrijalva/jwt-go v3.2.0+incompatible
github.com/emicklei/go-restful v2.12.0+incompatible
github.com/go-chassis/go-archaius v1.6.0-beta1
github.com/go-redis/redis v6.15.9+incompatible
github.com/go-redis/redis/v8 v8.1.0
github.com/golang/protobuf v1.5.2
github.com/google/pprof v0.0.0-20210720184732-4bb14d4b1be1
github.com/google/wire v0.5.0
github.com/hashicorp/golang-lru v0.5.4
github.com/jinzhu/copier v0.0.0-20190924061706-b57f9002281a
github.com/json-iterator/go v1.1.11
github.com/patrickmn/go-cache v2.1.0+incompatible
github.com/pkg/errors v0.9.1
github.com/satori/go.uuid v1.2.0
github.com/segmentio/kafka-go v0.3.4
github.com/spf13/cast v1.3.1
github.com/stretchr/testify v1.7.0
github.com/tealeg/xlsx v1.0.5
github.com/tidwall/gjson v1.8.1
github.com/vcaesar/murmur v0.20.1
github.com/vmihailenco/msgpack v4.0.4+incompatible
golang.org/x/net v0.0.0-20210726213435-c6fcb2dbf985
golang.org/x/oauth2 v0.0.0-20211104180415-d3ed0bb246c8
gonum.org/v1/plot v0.9.0
google.golang.org/api v0.68.0
google.golang.org/grpc v1.45.0
gopkg.in/go-playground/validator.v9 v9.29.1
gopkg.in/gomail.v2 v2.0.0-20160411212932-81ebce5c23df
gopkg.in/yaml.v2 v2.4.0
gorgonia.org/gorgonia v0.9.17
gorgonia.org/tensor v0.9.17
gorm.io/driver/mysql v1.1.1
gorm.io/gorm v1.21.12
```

执行curl命令将以上信息post到大促系统接口

- 大促系统维护组件白名单
- 大促系统-压测平台展示各服务组件信息，以及和组件白名单的差异

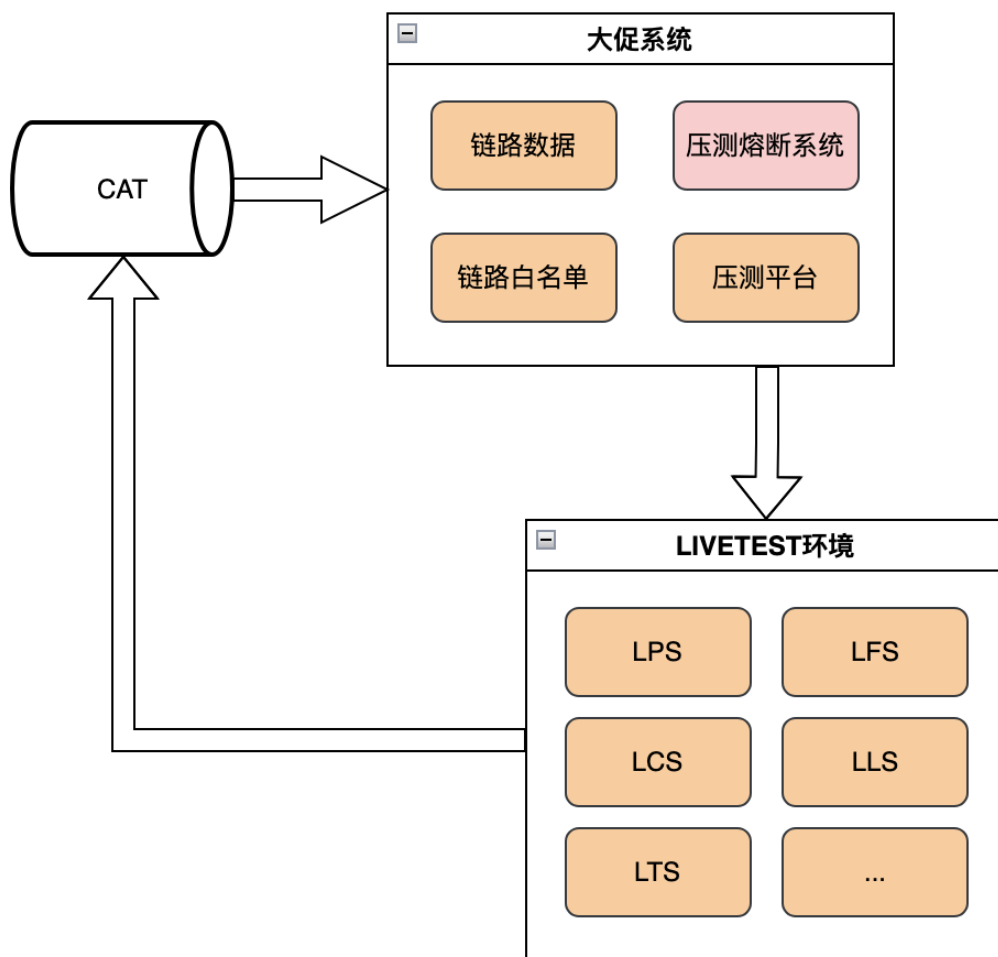
3.2 链路检查

压测链路以及影子流量可控是确保压测安全的关键。

随着代码迭代、架构升级，不断可能会有预期外的服务或中间件引入。

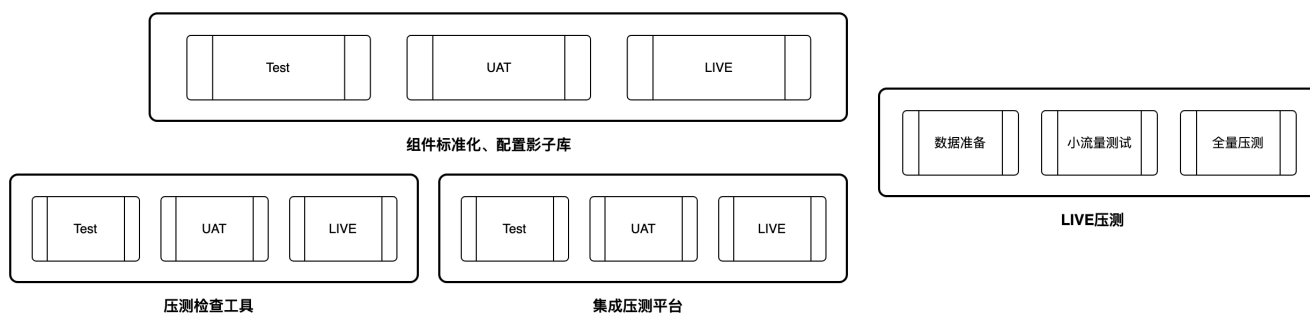
所以我们需要持续检查压测链路的情况，分析压测链路中的服务和中间件是否在白名单内。如发现预期外的链路出现，及早作出提示并熔断压测平台，直到压测链路符合预期。

因为live链路是抽样采集，live服务请求量比较大，不容易采集到我们发送的影子请求，我们选择在livetest环境采集和检查链路。



4 发布

4.1 上线方案



1. 各服务组件改造，并配置影子库，按照标准上线流程发布。
2. 同步进行检查工具开发，组件改造上线后启动检查流程。
3. 检查通过后，启动小流量测试、全量压测

4.2 资源需求

- 1.SLS履约链路涉及写入的影子库表。
- 2.SLS履约链路涉及写入的影子topic。
- 3.影子Redis集群（可选）。

[资源清单](#)

4.3 监控运维

1. CAT shadow domain
2. grafana配置shadow大盘

5 CheckList

功能测试、流量对比、UAT正常
检查CAT是否存在UnsafeContext上报
检查CAT中对应shadow domain上报是否符合预期
shadow日志是否符合预期
检查工具结果和人工检查结果对比
压测平台熔断功能是否正常
服务限流功能是否正常
压测数据是否符合预期
压测链路部署是否符合预期