美团大众点评微服务实践

-服务框架Pigeon的设计与实现

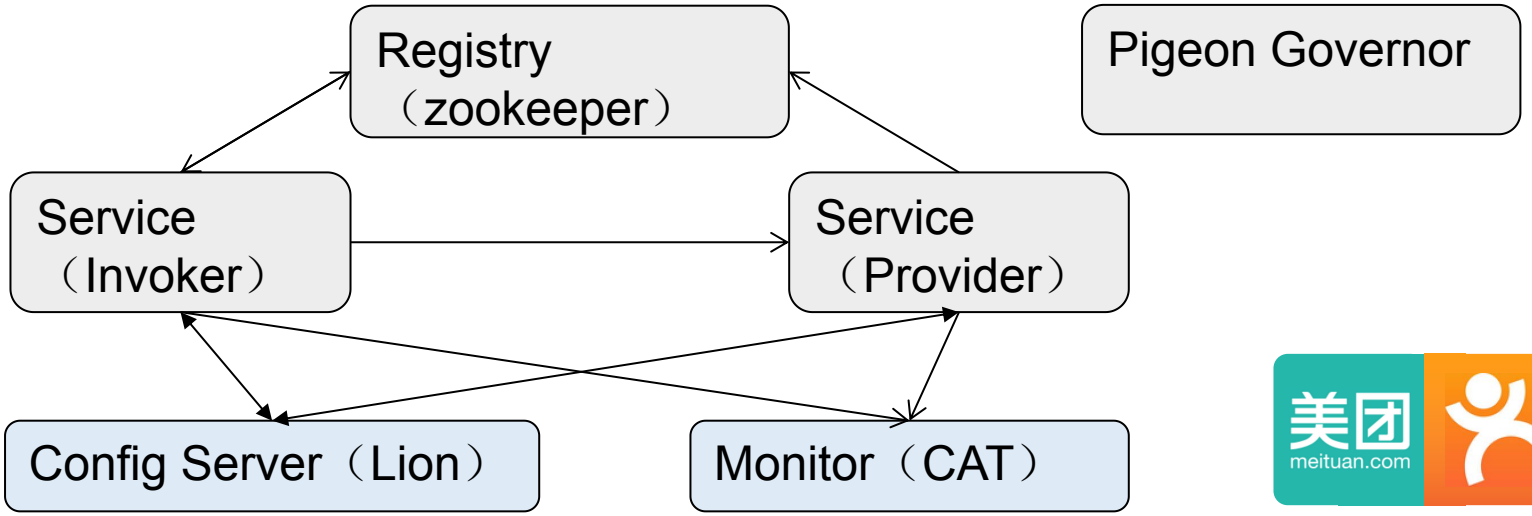
吴湘@美团大众点评，基础架构中心

大纲

- 服务框架Pigeon的设计与实现
- 服务治理
- 微服务的一些实践经验

服务框架Pigeon架构

- Pigeon提供jar包接入，线上运行在tomcat里
- Monitor-CAT，负责调用链路分析、异常监控告警等
- 配置中心-Lion，负责一些开关配置读取
- Governor-服务治理门户
- 一个interface定义为一个服务，每个服务有一个唯一标识



主要模块

服务治理门户

服务搜索	服务自助注册	服务治理分析
服务流控	服务降级	服务测试
运营报表	接口文档中心	

客户端

服务发现	服务调用	服务隔离
服务注册注销	监控分析	服务限流
服务心跳/重连	服务单机测试	服务降级
负载均衡	序列化	服务容错

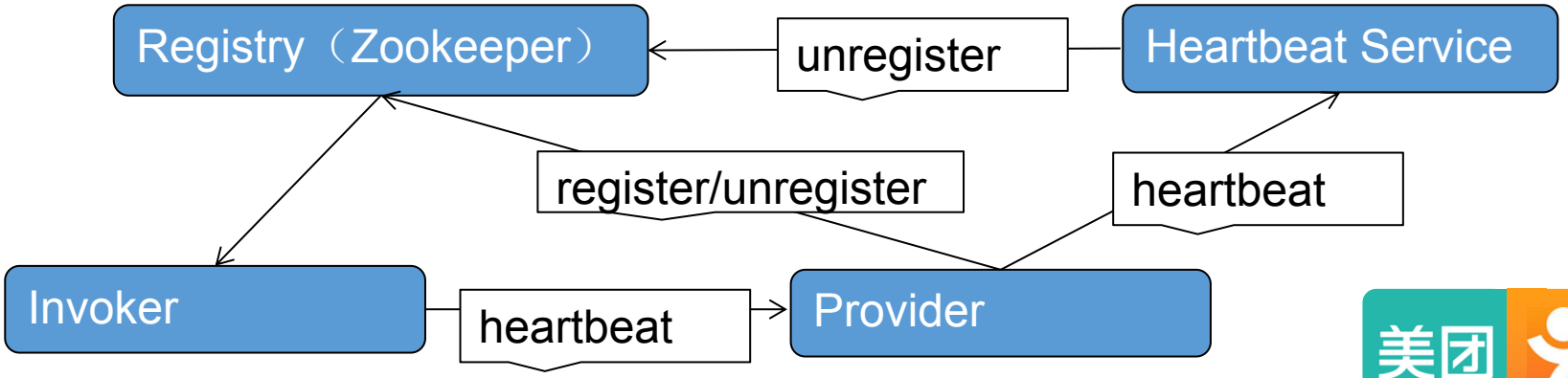
服务的注册与发现

- 注册信息包括service name、ip、port、group等
- 服务提供方初始化完成后自动注册，也可以通过api或管理端注册
- 服务调用方通过service name去发现服务

Registry（Zookeeper）
/DP/SERVER/com.dianping.iphub.IpService
10.66.1.1:5020,10.66.1.2:5020

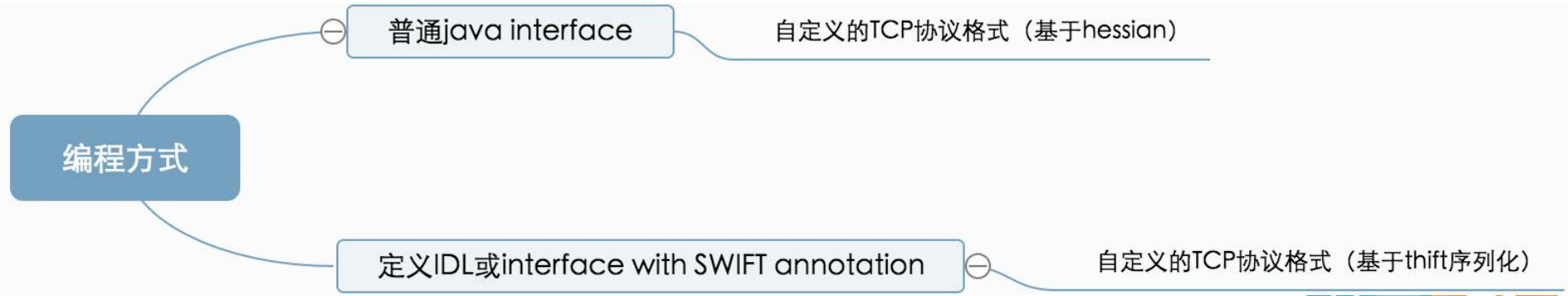
服务的注销

- 服务地址通过zookeeper持久节点存储，避免临时节点的不稳定
- 关闭tomcat时会调用pigeon脚本去注册中心摘除本机服务地址
- 对于残留的无效地址，有独立的心跳服务会检测无效的服务地址进行zookeeper删除
- 客户端对于无效的服务地址，内部也有心跳检测机制等来保证



编程方式、序列化

- 基于Hessian序列化，通过netty实现自定义TCP协议格式，开发成本低，通过java interface定义服务接口
- 基于Thrift序列化，通过netty实现自定义TCP协议格式，性能更高，开发成本稍高，通过定义IDL或annotation方式定义服务接口，更方便接入其他语言，thrift会有一些限制如方法不支持重载、struct不支持继承等

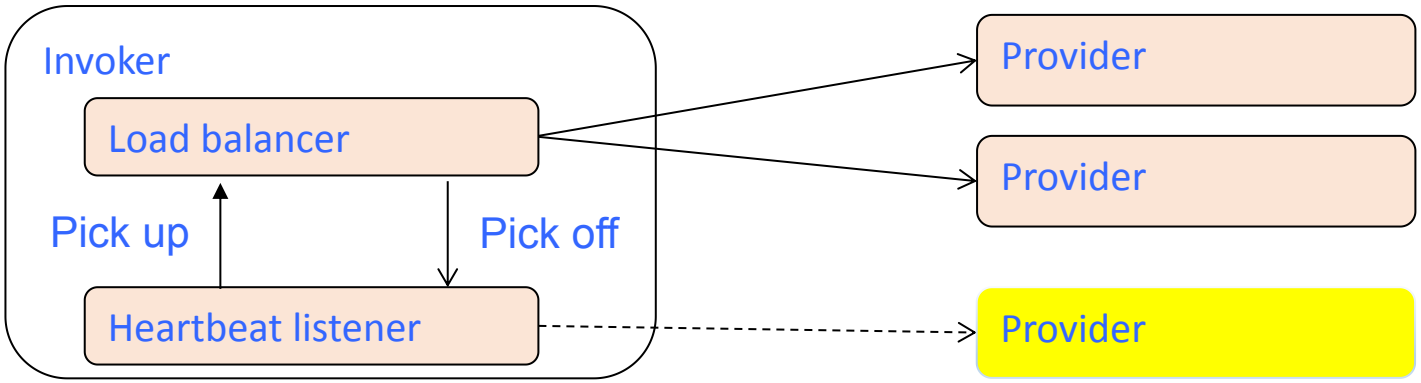


调用模式

- Sync，同步等待返回调用
- Future，可实现并行发出多个请求，总耗时是最慢的请求的执行时长，推荐方式
- Callback，发出结果不等待返回，结果回调，完全异步化
- Oneway，无需返回结果

客户端心跳

- 心跳线程客户端发起，定期发送，服务端响应，连续5次不成功则在本地路由缓存里摘除该服务端节点，摘除后下次尝试重连



客户端负载均衡

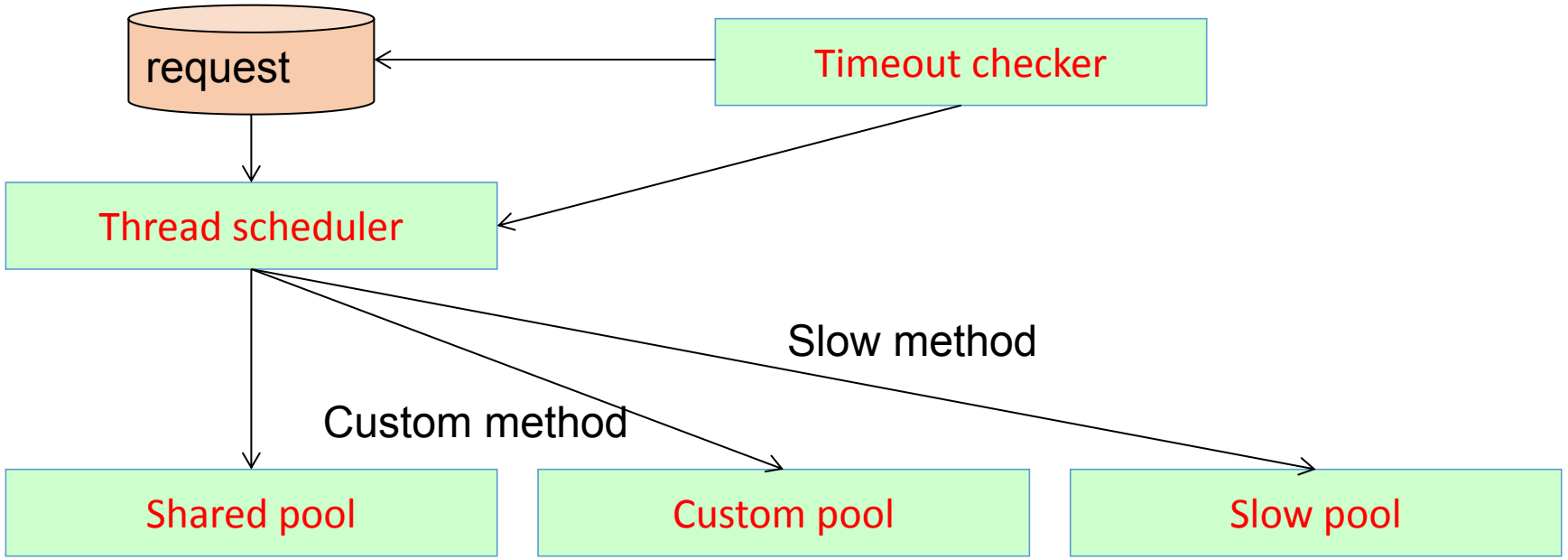
- 多种负载均衡策略，默认是自适应策略，客户端会计算发往每个服务端节点的在途请求数，新的请求会优先选择在途请求数最小的节点发送
- 预热控制，针对服务端某个刚启动的节点，客户端按从慢到快的频率，将请求逐步发往这个节点，防止服务端刚启动的节点大量请求进来导致大量超时
- 也可自定义负载均衡策略

服务限流

- 可以在服务端对某一个服务接口的某一个方法，针对不同的调用方应用的请求进行流量QPS限制，超出阀值后调用端会收到服务拒绝异常，未来会在调用端进行限流
- 服务端会对任意接口的请求所占用的线程数进行控制，防止单个接口某个方法用尽线程池所有可用线程

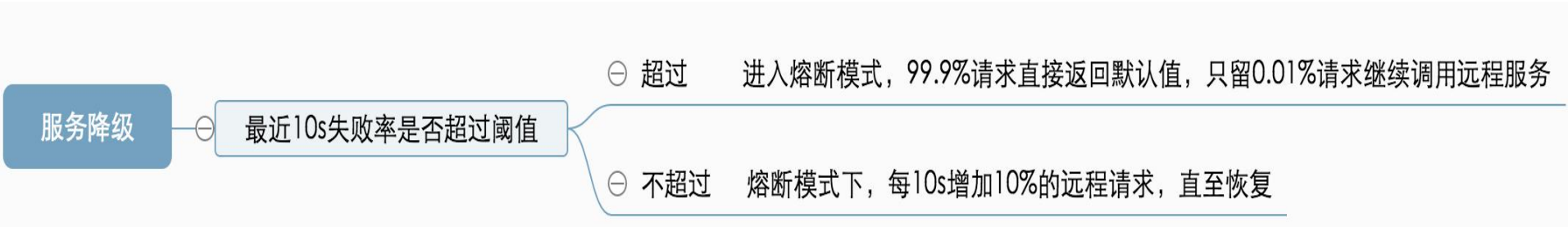
服务隔离

- 服务端默认会监控每个接口的超时情况，超时多的接口请求会自动路由到独立的慢线程池处理，如果该接口恢复正常，则会回到正常共享线程池处理
- 也可以为某些接口方法配置独立的线程池，剩余的使用公共池



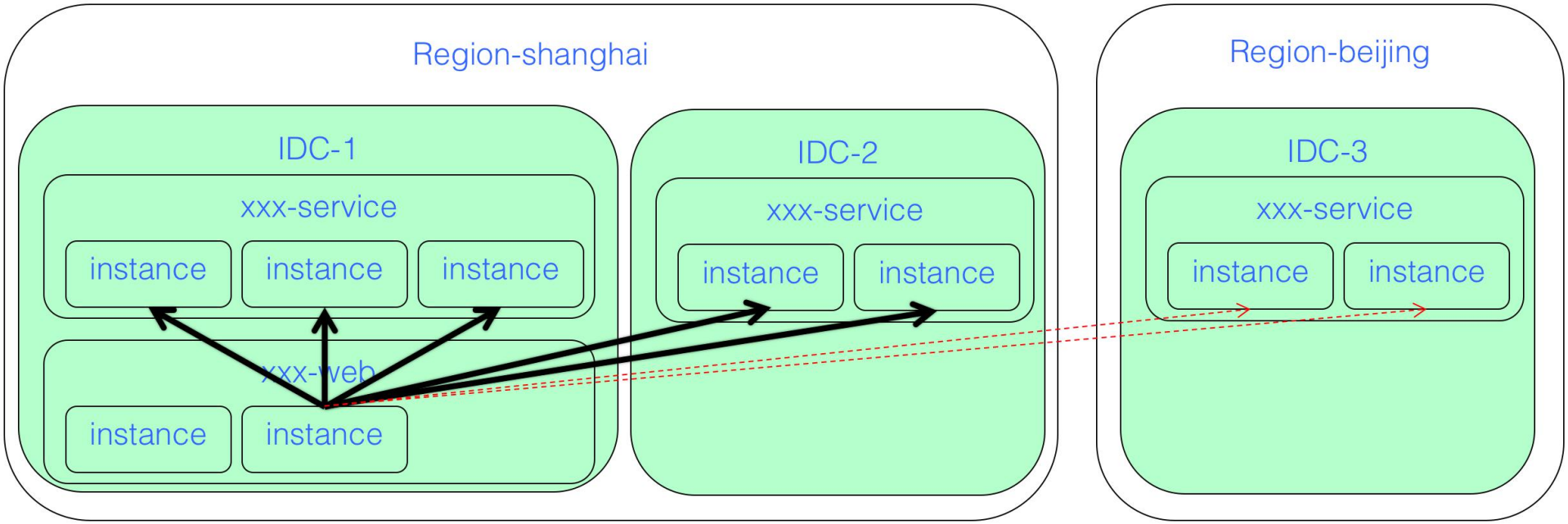
服务降级

- 若依赖的服务可以降级处理，Pigeon提供多种服务自动降级策略
- 降级的结果可以是自动返回默认值（支持json和groovy配置）、或抛出降级异常、或返回mock对象



多IDC支持

- 一个地域多个IDC，优先调用同地域的服务，也可配置优先调用同IDC的服务，同地域不同IDC可配比例

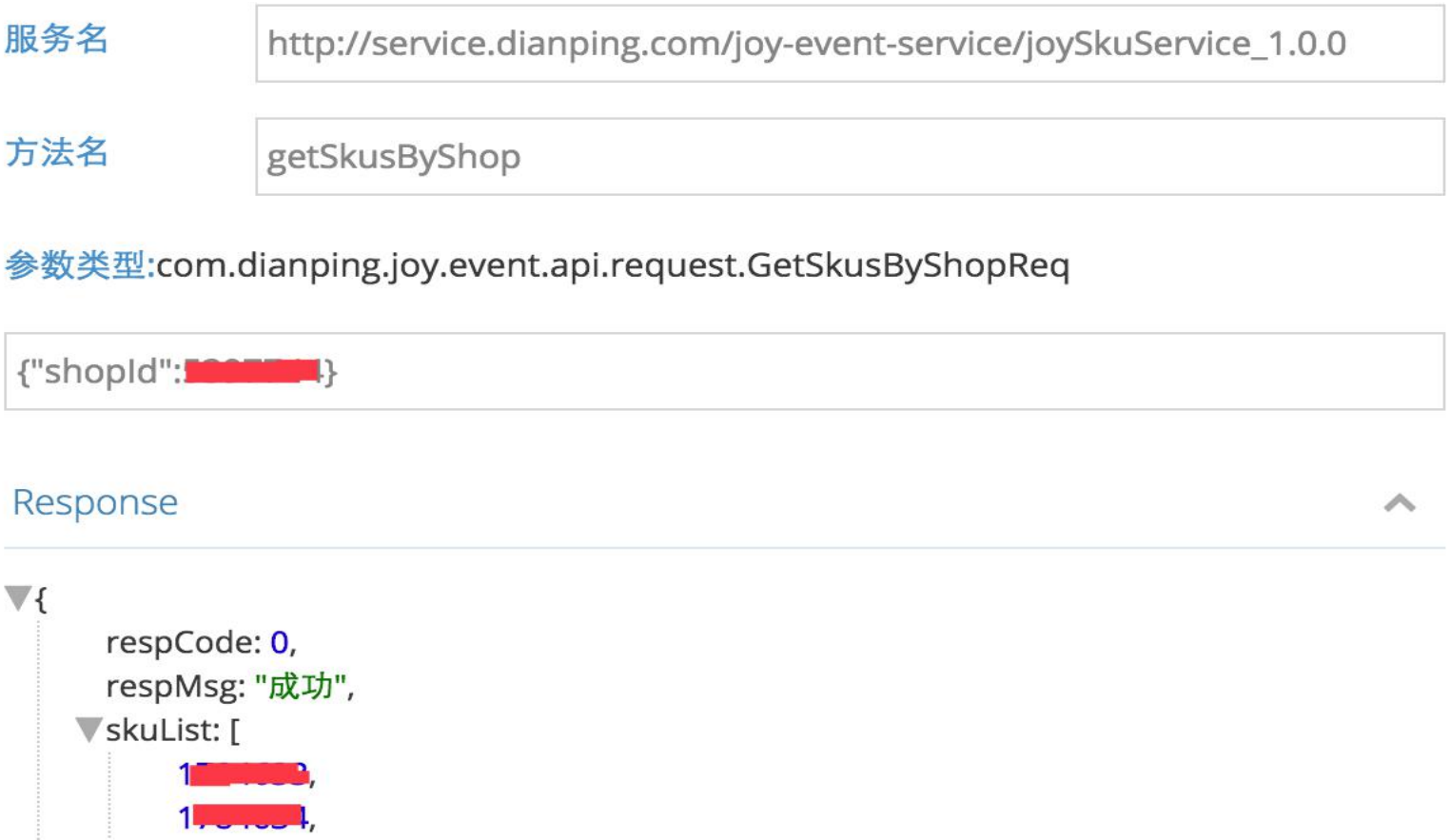


内置HTTP服务

- 内置4080端口的http服务，可以查看单机实时信息如QPS、注册状态、调用和被调实时状态、内部线程池状态等
- 通过http+（json/hessian）可以被其他应用通过GET或POST调用，未来会提供更好的REST服务
- 单机服务测试

服务测试

- 内置http服务，提供ip:4080/services服务测试页面，也可以通过管理端统一入口进行测试



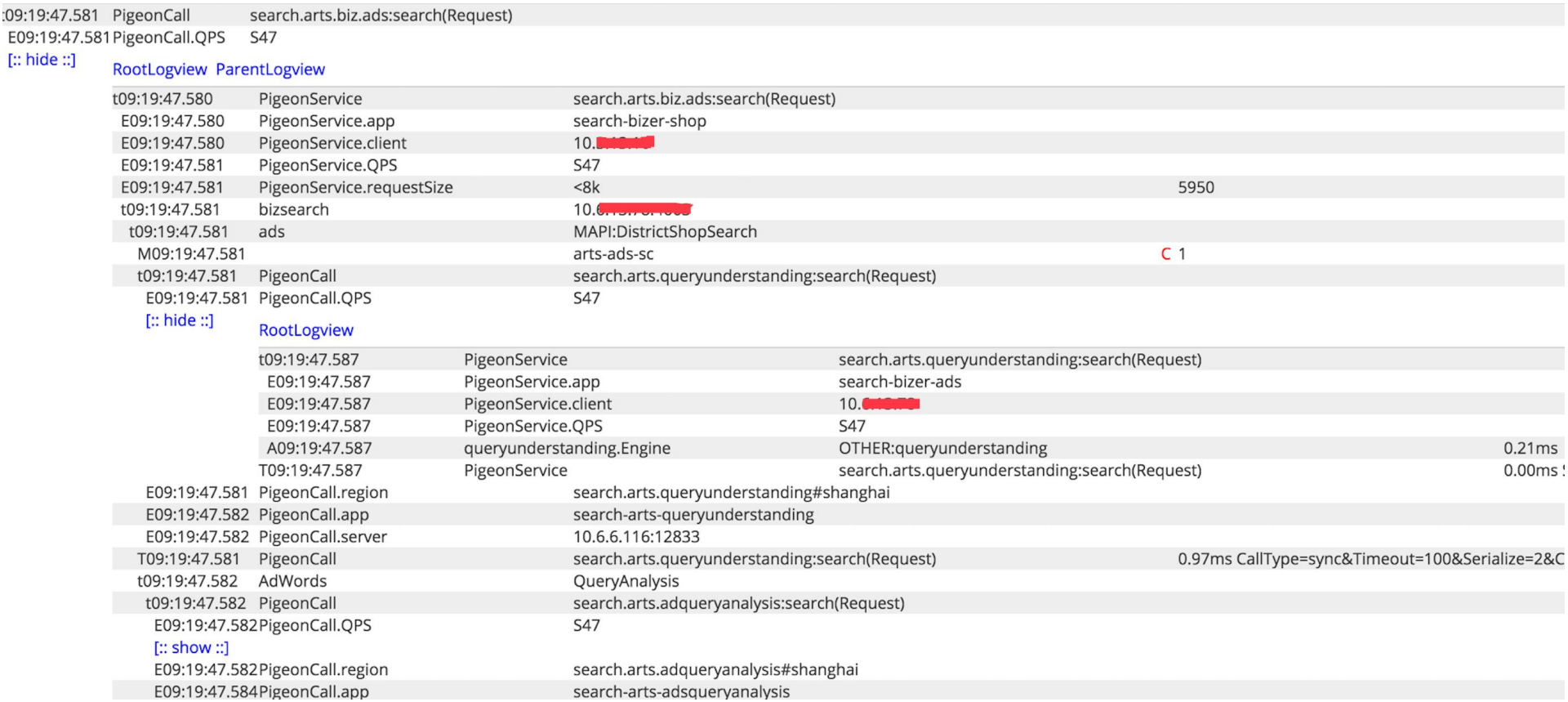
服务监控分析与告警

- 通过监控系统CAT（已开源）分析调用链路，包括调用量、TP耗时、异常、请求及响应大小、区间耗时明细、QPS等

Type	Total	Failure	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	95Line(ms)	99Line(ms)	99.9line(ms)	QPS
[:: show ::] Squirrel.client	257	0	0.0000%	Log View	2.8	290.5	42.6	71.0	71.0	71.0	0.1
[:: show ::] lion	14,972	0	0.0000%	Log View	0.5	14,526.5	23.2	26.1	35.7	4,366.2	4.3
[:: show ::] System	1,282	0	0.0000%	Log View	7.3	29.8	11.4	14.1	16.6	16.6	0.4
[:: show ::] PigeonCall	1,092,457	0	0.0000%	Log View	1	224	2.7	3.1	4.0	18.6	311.1
[:: show ::] PigeonService	19,338,129	0	0.0000%	Log View	0	254	0.4	1.0	1.0	1.9	5,506.3
[:: show ::] Squirrel.redis-coupon	20,378,251	969	0.0048%	Log View	0	169	0.2	0.0	0.0	0.9	5,802.4

Type	Total	Status	Count	SampleLinks
error [:: show ::]	4,400	[:: show ::] com.dianping.pigeon.remoting.invoker.exception.RequestTimeoutException	4,379	Loooooooooooooooooooooooooooooooooooooooooog
		[:: show ::] com.dianping.pigeon.remoting.provider.exception.ProcessTimeoutException	21	Looooooooooooooooooog

服务监控分析与告警



:09:19:47.581	PigeonCall	search.arts.biz.ads:search(Request)		
E09:19:47.581	PigeonCall.QPS	S47		
[:: hide ::]	RootLogview ParentLogview			
t09:19:47.580	PigeonService	search.arts.biz.ads:search(Request)		
E09:19:47.580	PigeonService.app	search-bizer-shop		
E09:19:47.580	PigeonService.client	10.[redacted]		
E09:19:47.581	PigeonService.QPS	S47		
E09:19:47.581	PigeonService.requestSize	<8k	5950	
t09:19:47.581	bizsearch	10.[redacted]		
t09:19:47.581	ads	MAPI:DistrictShopSearch		
M09:19:47.581		arts-ads-sc	C 1	
t09:19:47.581	PigeonCall	search.arts.queryunderstanding:search(Request)		
E09:19:47.581	PigeonCall.QPS	S47		
[:: hide ::]	RootLogview			
t09:19:47.587	PigeonService	search.arts.queryunderstanding:search(Request)		
E09:19:47.587	PigeonService.app	search-bizer-ads		
E09:19:47.587	PigeonService.client	10.[redacted]		
E09:19:47.587	PigeonService.QPS	S47		
A09:19:47.587	queryunderstanding.Engine	OTHER:queryunderstanding		0.21ms
T09:19:47.587	PigeonService	search.arts.queryunderstanding:search(Request)		0.00ms
E09:19:47.581	PigeonCall.region	search.arts.queryunderstanding#shanghai		
E09:19:47.582	PigeonCall.app	search-arts-queryunderstanding		
E09:19:47.582	PigeonCall.server	10.6.6.116:12833		
T09:19:47.581	PigeonCall	search.arts.queryunderstanding:search(Request)	0.97ms CallType=sync&Timeout=100&Serialize=2&C	
t09:19:47.582	AdWords	QueryAnalysis		
t09:19:47.582	PigeonCall	search.arts.adqueryanalysis:search(Request)		
E09:19:47.582	PigeonCall.QPS	S47		
[:: show ::]				
E09:19:47.582	PigeonCall.region	search.arts.adqueryanalysis#shanghai		
E09:19:47.584	PigeonCall.app	search-arts-adsqueryanalysis		

服务治理

- 服务可用性、耗时（平均、TP99等）排行运营日报
- 调用深度过长（>4）统计
- 出度，入度过大的服务、过大的服务统计
- 过长的超时时间统计
- 检测循环调用的风险
- 检测可能可以并行调用的服务
- 梳理核心服务性能冗余度，基础底层服务建议采用全异步化等手段提高吞吐

微服务的一些实践-基础设施标准化

- 标准化运行环境，如无特殊情况，线上业务的虚拟机KVM或docker配置一样
- 标准化运行容器如Tomcat
- 标准化环境标识，如每台机器固定路径的appenv文件，env=production，文件内容标识机器属于什么环境如线上环境、测试环境
- 标准化应用名称规范，每个应用有一个唯一的名称，如war包下放置一个app.properties，app.name=xxx-xxx-service

微服务的一些实践-基础设施标准化

- 统一的开发语言
- 标准化发布工具，如可以实现统一war包发布、jar包版本升级限制
- 统一的服务通信框架
- 统一的配置中心
- 统一的数据库、KV等存储访问层
- 统一的MQ
- 统一的监控系统

微服务的一些实践

- 底层存储如mysql、KV等尽量保证只有一个或少数几个服务访问，每个应用只能访问自己的存储
- 面向业务领域定义服务（interface），每个服务高内聚，一个应用可能多个服务
- 按业务产品线规划应用，理解业务本质，根据业务发展情况进行服务的拆分或重构
- 组织结构按业务产品线划分，做好微服务需组织结构支持

Q/A

开源地址及联系方式

- pigeon开源地址：https://github.com/dianping/pigeon
- 个人联系方式：微信wux_china