



Trouble Shooting In Action

--线上问题的快速定位与解决

2018-04 陈金

可用性



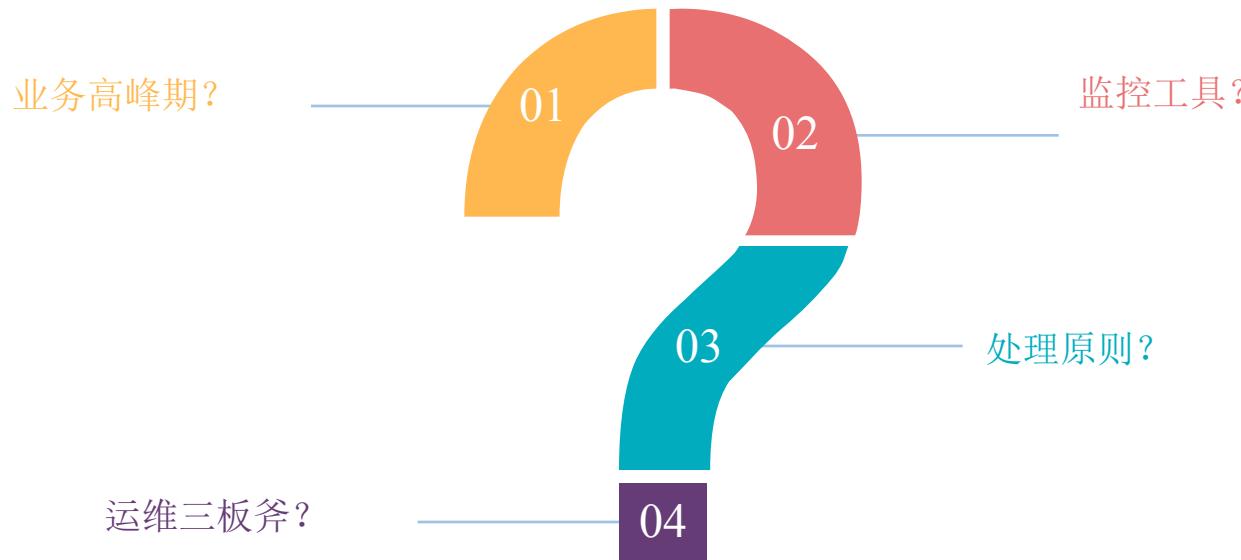
可用性	全年不可用时间
1个9	$365 * (1 - 90\%) = 36.5$ 天
2个9	$365 * (1 - 99\%) = 3.65$ 天
3个9	$365 * 24 * (1 - 99.9\%) = 8.76$ 小时
4个9	$365 * 24 * 60 * (1 - 99.99\%) = 52.56$ 分
5个9	$365 * 24 * 60 * (1 - 99.999\%) = 5.256$ 分
6个9	$365 * 24 * 60 * 60 * (1 - 99.9999\%) = 31.5$ 秒

点餐Q1故障平均处理时间

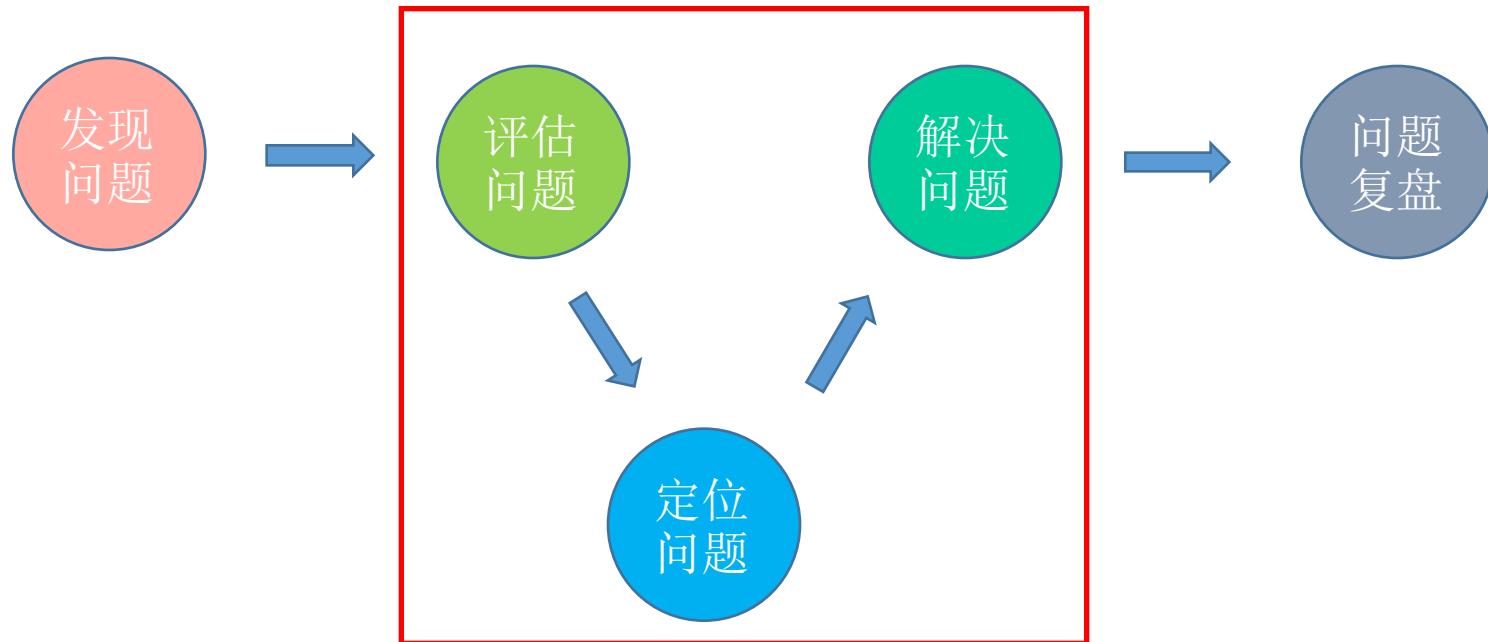


美团点评

故障名称	持续时间
取餐码改造导致DDT等渠道不可用	7小时26分钟
项目迁移SLB配置错误导致接单失败	1小时10分钟
info-bom升级导致门店管理、菜品管理不可用	1小时15分钟
菜单页菜品展示不全	1小时18分钟
部分IOS系统 -小程序无法下单	2小时11分钟
基础版老服务下线导致无法下单	9分钟



线上问题处理流程





美团点评



01

业务理解

02

知识&工具

03

原则&手段

04

案例分析

05

方法论沉淀

01

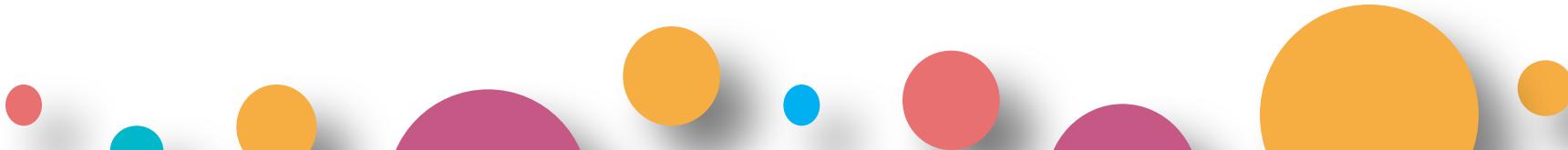


业务理解

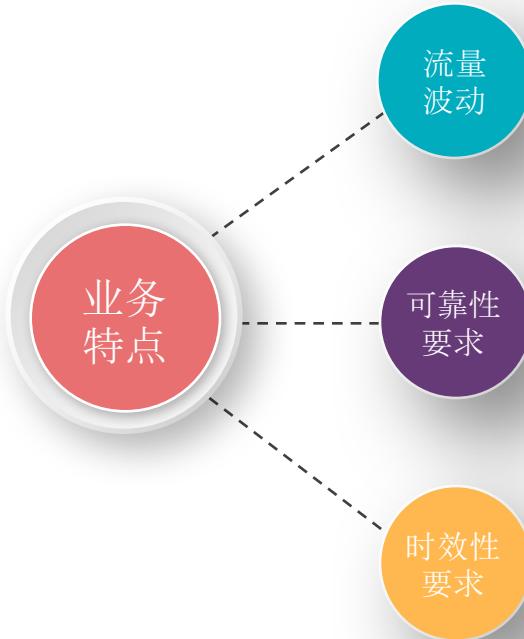
业务特点

业务流程

核心链路



连接用户与商家



午市&晚市高峰。周末节假日小高峰。

商家可能无纸质菜单；下单到不了厨房体验糟。

不同于电商，履约时效性高，用户下单需快速给反馈

业务高峰期

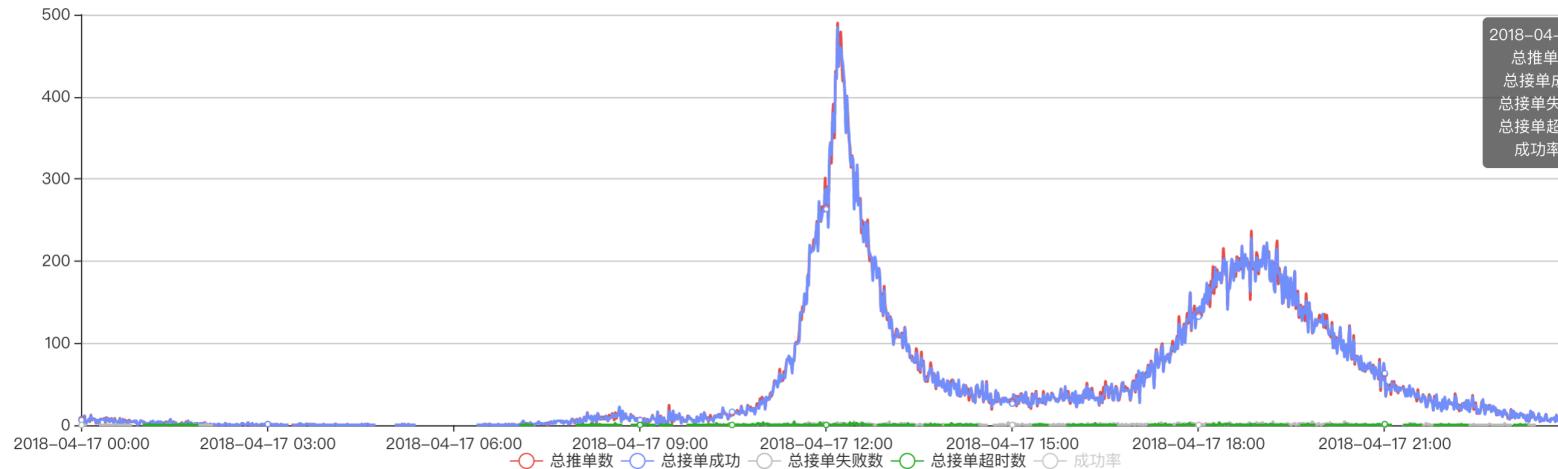


接单总体情况

2018-04-17 00时 至 2018-04-18 00时



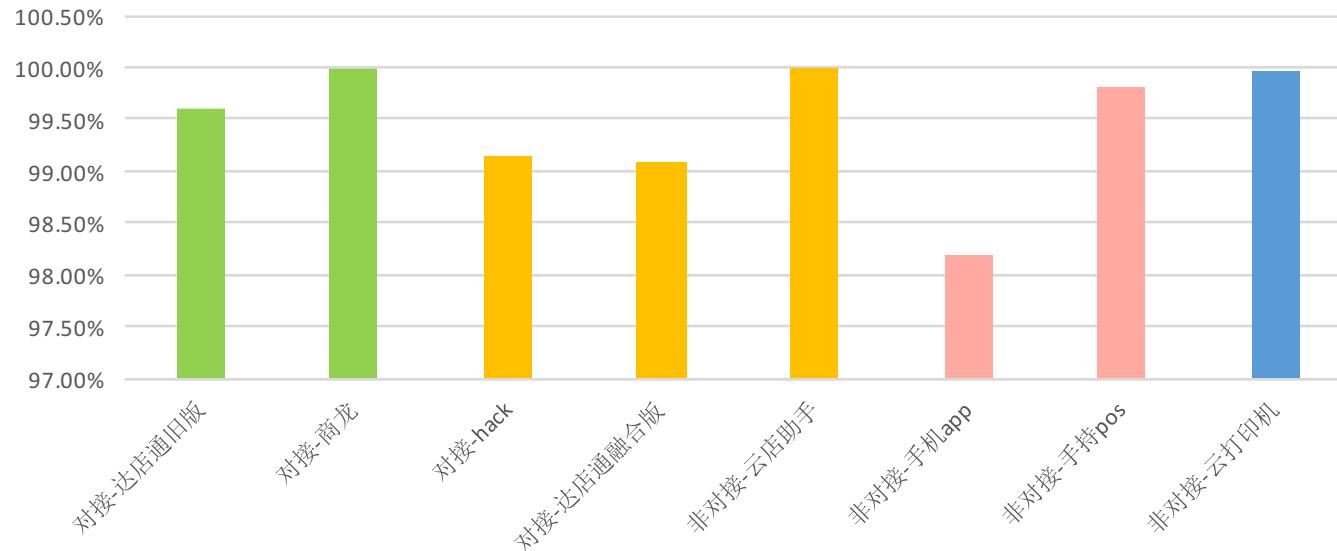
2018-04-17 10:29
总推单数: 15
总接单成功: 16
总接单失败数: 0
总接单超时数: 0
成功率: 100%



接单成功率



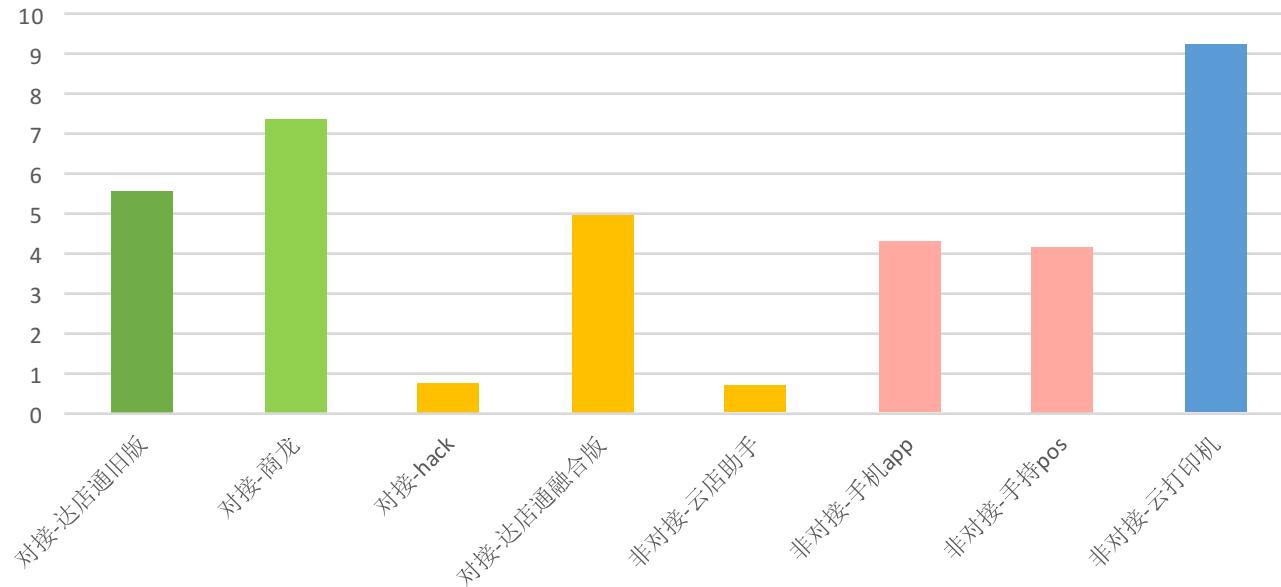
成功率



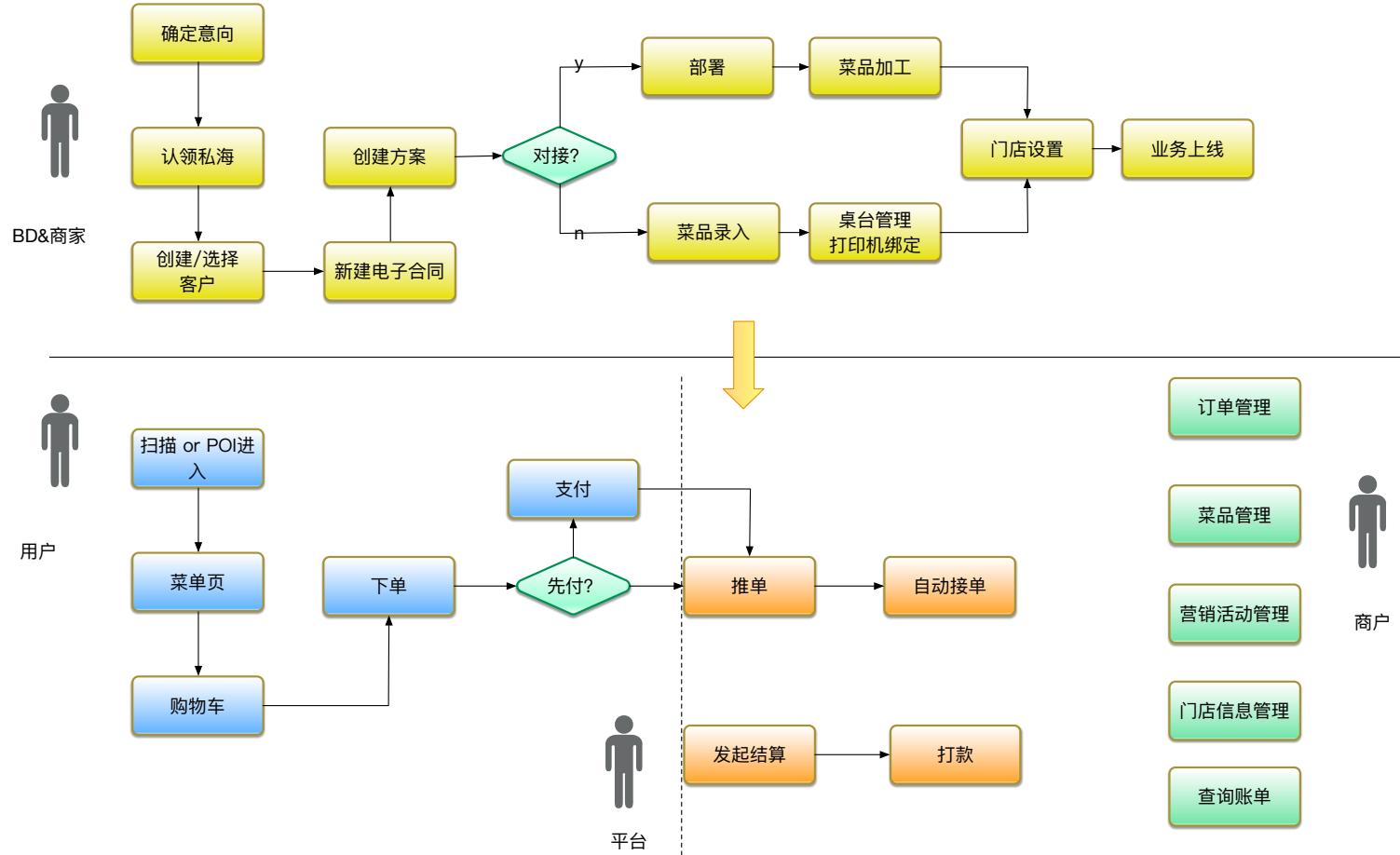
接单时长



接单时长(s)



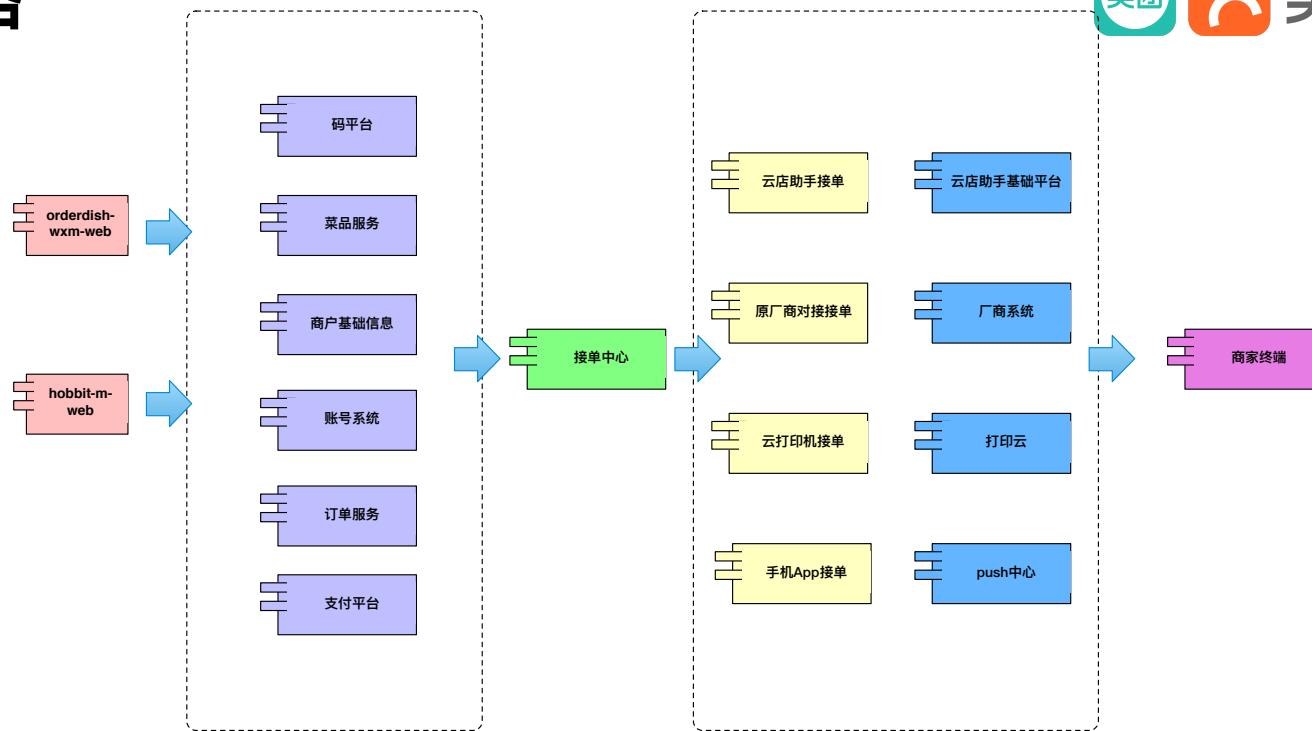
业务流程



核心链路



美团点评



redis-orderdish/redis-orderdish-shop/
redis-orderdish-platform/etc



orderdishrecord/orderdishpay/orderdishuser/
Orderdishmenu/orderdishshopbase/etc



02

知识&工具

部署结构

监控体系

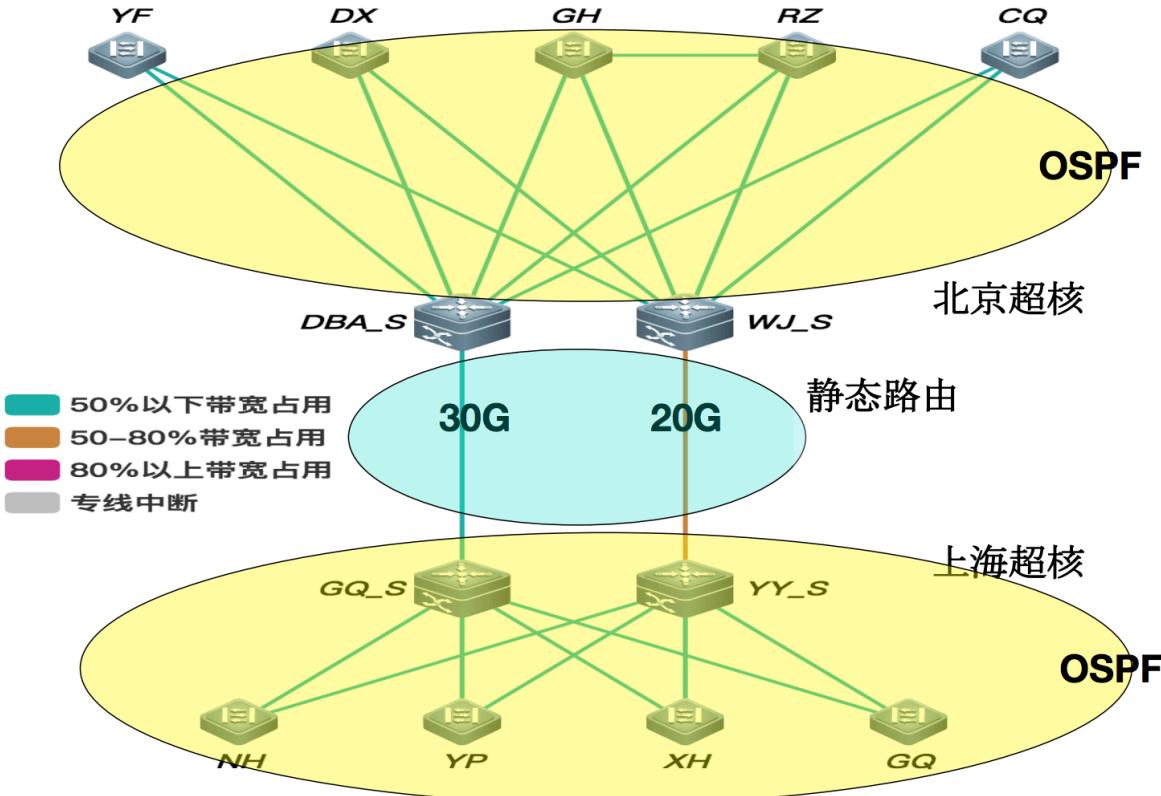
专业知识

中间件与运维工具

网络拓扑



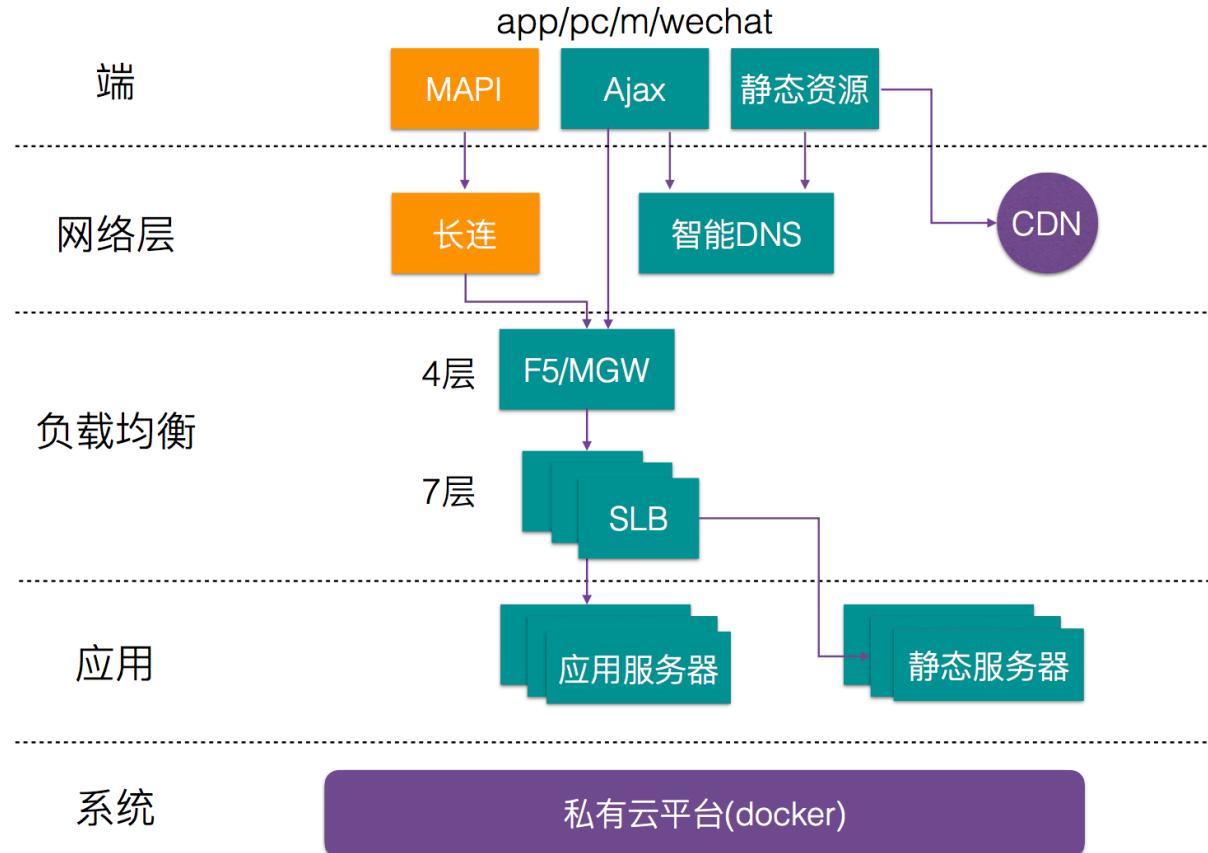
美团点评



部署结构



美团点评



应用机器与容器



机器信息

- cpu核心
- 内存
- 磁盘
- 网卡

生产	3	IP	泳道	类型	状态	CPU	内存	机房	上线时间	web终端	更多
		orderdish-ecom-web01.yp	10.72.32.227	default	虚拟机	在线	4	4GB	月浦	2017-03-31 10:33:03	<button>登录</button> <button>详情</button>
		orderdish-ecom-web02.yp	10.72.33.139	default	虚拟机	在线	4	4GB	月浦	2017-03-31 10:33:05	<button>登录</button> <button>详情</button>
		orderdish-ecom-web01.gq	10.69.211.80	default	虚拟机	在线	4	8GB	桂桥	2017-09-14 11:00:44	<button>登录</button> <button>详情</button>

关键配置

- /usr/local/nginx/conf/nginx.conf
- /usr/local/tomcat/conf/server.xml

```
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
          maxThreads="300" maxQueueSize="80" minSpareThreads="30" maxIdleTime="60000"/>
  <Connector executor="tomcatThreadPool" port="8000" protocol="org.apache.coyote.http11.Http11NioProtocol"
    URIEncoding="UTF-8" compression="off" enableLookups="false" maxKeepAliveRequests="20" bufferSize="8192"
connectionTimeout="5000"
    redirectPort="8443" maxPostSize="20971520" server="DPServer"/>
```

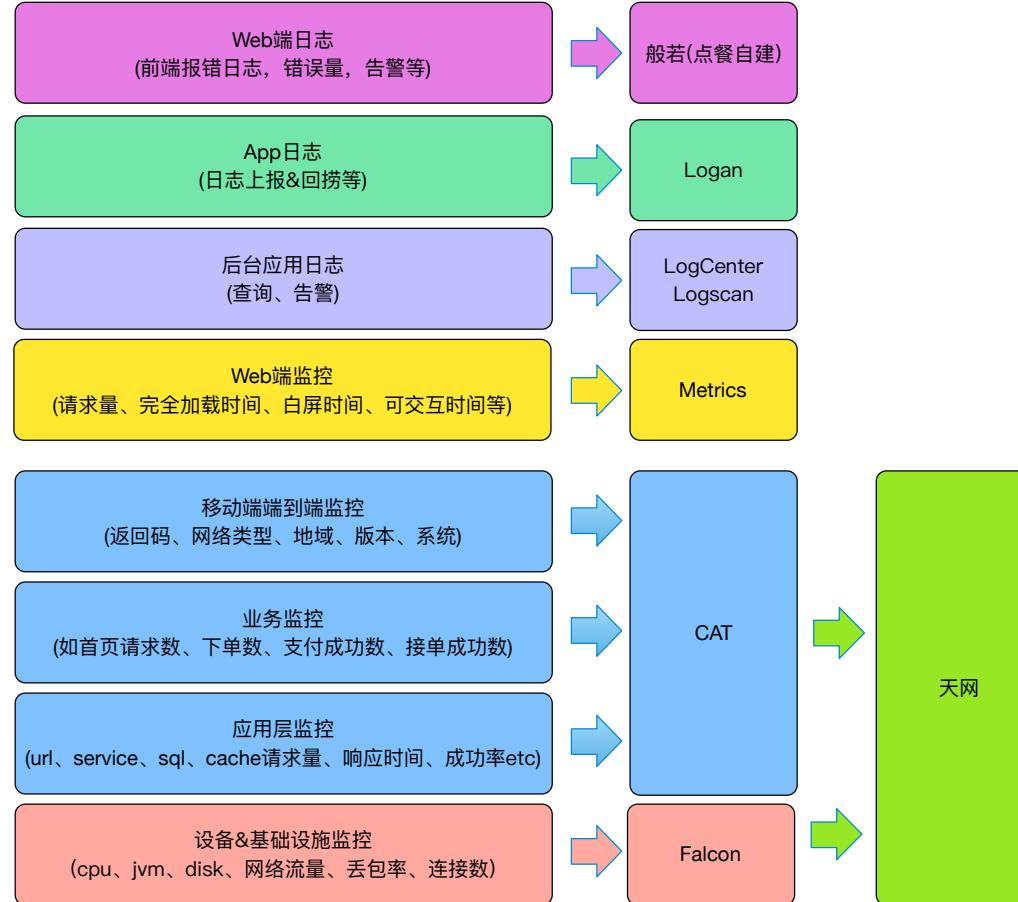
connection_pool_size	256;
request_pool_size	64k;
output_buffers	2 128k;
postpone_output	1460;
client_header_timeout	1m;
client_body_timeout	3m;
send_timeout	3m;
log_not_found	on;
sendfile	on;
tcp_nodelay	on;
tcp_nopush	off;
reset_timedout_connection	on;
keepalive_timeout	10 5;
keepalive_requests	100;



美团点评

监控体系

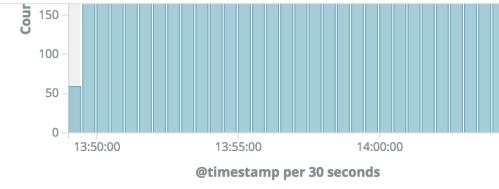
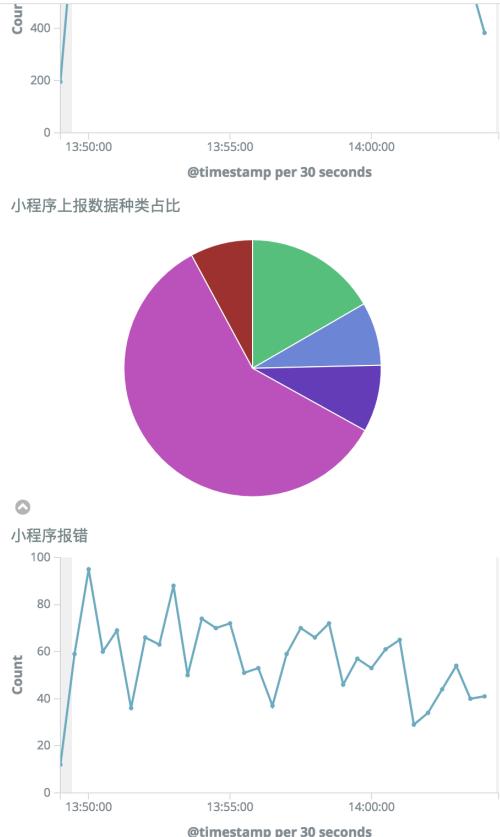
日志&监控



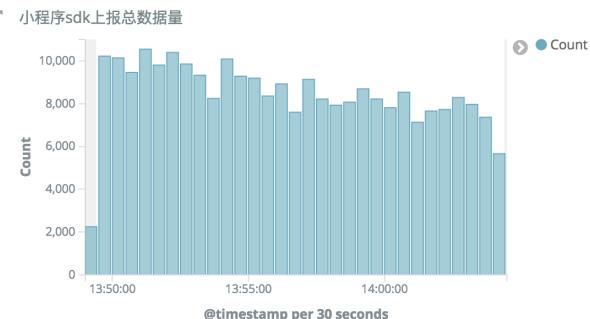
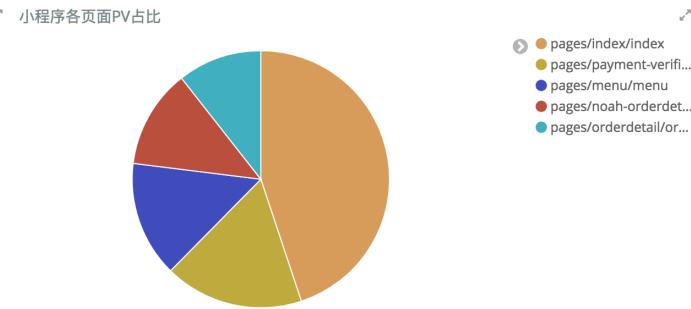
kibana

- Discover
- Visualize
- Dashboard**
- Timeline
- Management

Collapse



- xhr
- pv
- pd
- moduleView
- moduleClick



L, like Gecko Chrome/49.0.2623.87 Safari/537.36 userAgent.browser: Google Chrome 49.0.2623.87

Metrics



美团点评

Metrics

总览

页面

浏览器

操作系统

地理区域

Ajax

指标查询

浏览器基本指标

浏览器自定义指标 (新)

全站性能

浏览量 (次)

709

完全加载时间 (统计方法: TP90)

4.9 s

可交互时间 (统计方法: TP90)

2.2 s

首字节时间 (统计方法: TP90)

655 ms

Ajax性能

调用量 (次)

3669

出错量 (次)

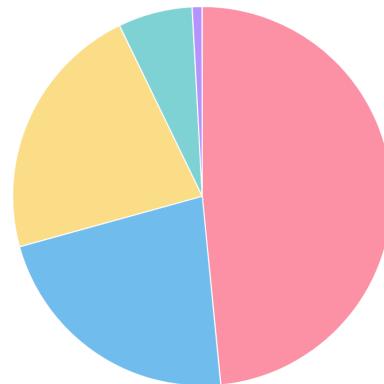
13

响应时间 (统计方法: TP90)

693 ms

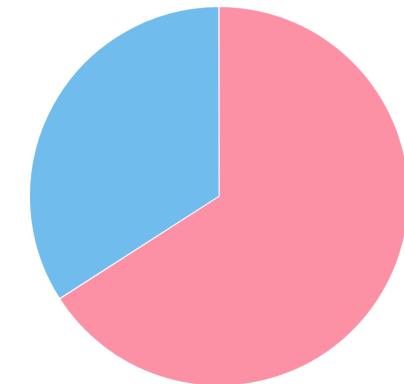
浏览器请求量 (次)

- iOS Webview (WebKit)
- Chrome WebView
- WeChat
- TitansX (default)
- Android Browser



操作系统占比分布 (次)

- iOS
- Android



<http://metrics.sankuai.com/browser/app-menuorder-h5/specification/page>

Logan



欢迎你, 陈金 [退出](#)

全选 Logan内部日志 用户行为日志 代码级日志 Shark日志 端到端日志 PushSdk日志 点评崩溃日志 灵犀日志 FatCat日志 [×](#)

外卖日志 SharkPush luban日志 SandBox日志 h5日志 owl日志 前端redux日志

时间	日志类型	日志简介
01:25:25.390	灵犀日志	页面加载:[页面:history_deal]
01:25:25.074	灵犀日志	页面加载:[页面:history_shop]
01:25:25.060	灵犀日志	页面关闭:[页面:NVMainViewController]
01:25:24.827	灵犀日志	业务上报:[点击][打点:delete][页面:history_shop]
01:25:24.683	端到端日志	(200) tunnel_big_request 168ms
01:25:24.682	端到端日志	(200) mapi.dianping.com/mapi/recentlyviewshop.bin 170ms
01:25:24.512	灵犀日志	页面加载:[页面:history_shop]
01:25:24.483	灵犀日志	业务上报:[点击][打点:history][页面:me]
01:25:23.535	灵犀日志	页面加载:[页面:me]
01:25:23.519	灵犀日志	页面关闭:[页面:NVShopHistoryViewController]
01:25:23.137	端到端日志	(200) mapi.dianping.com/mapi/usercenter/uniordercount.bin 112ms
01:25:23.118	端到端日志	(200) mapi.dianping.com/mapi/framework/getelements.api 96ms
01:25:22.999	灵犀日志	业务上报:[点击][打点:back][页面:history_shop]
01:25:20.174	灵犀日志	业务上报:[点击][打点:edit][页面:history_shop]
01:25:19.328	端到端日志	(200) _pic_https://qcloud.dpfile.com/pc/v8tuJuYiUyh5X143-GMoVrRXWNQTdfmZ8S9jSeeM...
01:25:19.047	端到端日志	(200) _pic_https://img.meituan.net/msmerchant/1cb61cc6b31648f8bc5e08d1d09a428c2...

日志详情

日志类型 : 代码级日志
时间(同步) : 2018-04-19 01:24:35.898
线程名称 :
线程ID : 1
是否主线程 : 是

时间戳(本地) : 01:24:35.898
日志分类 : 一般
调用位置 : NVPushGuideHelper.m 102
日志内容 : <NVPushGuideHelper.m : line 102> func:-[NVPushGuideHelper userActionRequest:]_block_invoke log:NVPushGuideHelper 上报成功:ok content:{
"scene_type" : "1",
"search_key" : "ing搏击健身工作室",
"search_tag" : "1",
"action_type" : "1"
}

原始日志 :

```
{  
    "timestamp": 1524072275898.154,  
    "level": "normal",  
    "log": "<NVPushGuideHelper.m : line 102> func:-[NVPushGuideHe
```

<https://logan.sankuai.com/logan/automatismUpload.html?appId=1>

CAT (Central Application Tracking)

Application Mobile Frontend Configs HOME

需求收集 Star 环境切换 您好 kevin.chen

Dashboard SLA Transaction Event Problem Heartbeat Business Dependency RPC API Service Database Cache Search MQqueue Logview State

2018-04-20 17:00 ~ 2018-04-20 17:59 常用 mp-official-menu-service Go 【切到历史】 [-1d] [-1h] [+1h] [+1d] [now]

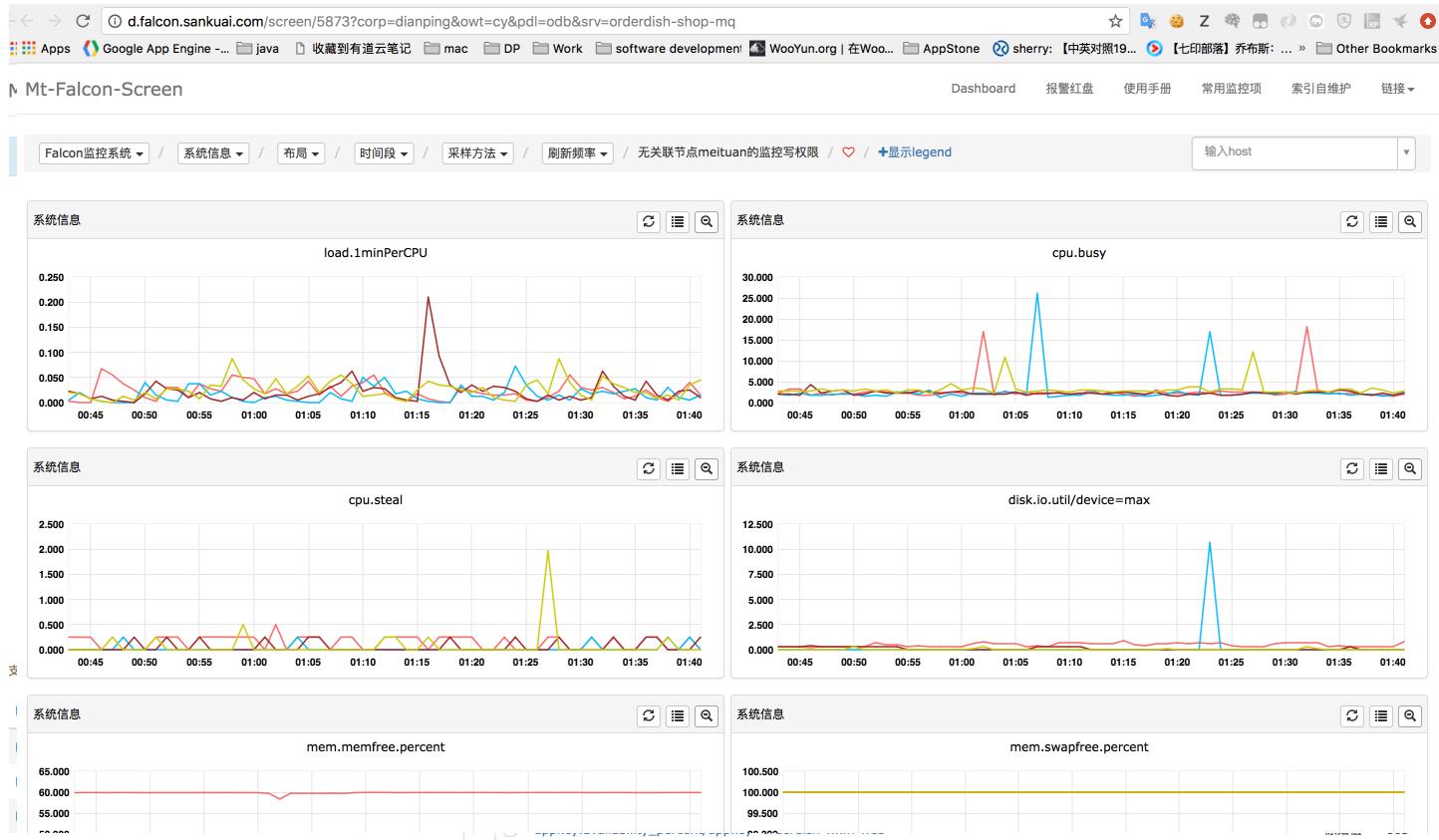
[All] 搜索 回车搜索 机器列表

[10.69.113.241 (mp-official-menu-service01.gq)] [10.69.113.235 (mp-official-menu-service02.gq)] [10.72.183.178 (mp-official-menu-service02.yq)] [10.69.211.125 (mp-official-menu-service03.gq)]

Long-url 1000 ms Long-sql 1000 ms Long-service 1000 ms Long-cache 25 ms Long-call 100 ms 查询 分钟级分布 自定义阈值 开始 0 结束 59 (分布饼图)

Type	Total	Status	Count	SampleLinks
error	2	[: show ::] org.springframework.dao.DataIntegrityViolationException	1	L
		[: show ::] com.mysql.jdbc.MySQLDataTruncation	1	L
SQL	1	[: show ::] OperationLogMapper.insert	1	L
long-cac he	4	[: show ::] HasSetMealCache-NEW:get	3	Log
		[: show ::] hack-sku-sold-out:get	1	L
long-ser vice	1	[: show ::] spu:skuService_1.0.0:updateAll(List,MenuCont ext)	1	L
Heartbe at	168	[: show ::] 10.69.113.235 system status	42	Looooooooooooooooooooog
		[: show ::] 10.69.211.125 jvm	42	Looooooooooooooooooooog
		[: show ::] 10.72.183.178 jstack	42	Looooooooooooooooooooog
		[: show ::] 10.69.113.241 etc.	42	Looooooooooooooooooooog

Falcon



<http://d.falcon.sankuai.com/>

<http://s.falcon.sankuai.com/screen/>

<https://tw.sankuai.com/>



美团点评

专业知识

专业知识



CPU、内存、磁盘、etc.



协议栈、DNS、Cookie、etc.



索引、锁、事务、etc.



多线程、锁机制、GC机制、etc.

CPU



美团点评

CPU核数: cat /proc/cpuinfo

运行状态: top

- load average: queued + runnable
- us time: user cpu time
- sy time: system cpu time
- ni time: user nice cpu time
- wa time: io wait cpu time
- hi time: hardware irq
- st time: software irq
- id time: idle cpu time

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10781	nobody	20	0	5410m	2.4g	6168	S	28.3	67.2	639:40.95	java
11576	phoenix	20	0	2601m	254m	9508	S	1.0	6.9	181:14.82	java
23877	nobody	20	0	85592	22m	812	S	1.0	0.6	2:10.24	nginx
23878	nobody	20	0	85528	22m	812	S	0.7	0.6	2:04.19	nginx
23879	nobody	20	0	85716	22m	812	S	0.7	0.6	2:10.96	nginx
6267	nobody	20	0	882m	12m	1972	S	0.3	0.3	16:11.74	sg_agent
8394	yunshuan	20	0	19192	1324	1008	R	0.3	0.0	0:00.60	top
10391	root	20	0	629m	10m	2488	S	0.3	0.3	47:18.20	falcon-agent
1	root	20	0	19232	884	592	S	0.0	0.0	0:01.15	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0		migration/0
4	root	20	0	0	0	0	S	0.0	0.0		irqd/0
5	root	RT	0	0	0	0	S	0.0	0.0		migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:02.29	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:05.78	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	0:35.36	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:01.27	watchdog/1

进程统计信息

进程信息

内存使用量

使用率

线上机器jvm参数： ps -ef | grep 'java'

```
-Xms5114m -Xmx5114m -Xss512k -XX:PermSize=350m -XX:MaxPermSize=350m -XX:NewSize=1967m -XX:MaxNewSize=1967m  
-XX:SurvivorRatio=8 -XX:MaxTenuringThreshold=9 -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly  
-XX:+CMSScavengeBeforeRemark -XX:+ScavengeBeforeFullGC -XX:+UseCMSCompactAtFullCollection -XX:+CMSParallelRemarkEnabled  
-XX:CMSFullGCsBeforeCompaction=9 -XX:CMSInitiatingOccupancyFraction=80 -XX:+CMSClassUnloadingEnabled  
-XX:SoftRefLRUPolicyMSPerMB=0 -XX:-ReduceInitialCardMarks -XX:+CMSPermGenSweepingEnabled  
-XX:CMSInitiatingPermOccupancyFraction=80 -XX:+ExplicitGCI invokesConcurrent  
-Djava.nio.channels.spi.SelectorProvider=sun.nio.ch.EPollSelectorProvider  
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -XX:+PrintGCDetails -XX:+PrintGCDateStamps  
-XX:+PrintGCA pplicationConcurrentTime -XX:+PrintGCA pplicationStoppedTime -XX:+PrintHeapAtGC -Xloggc:/data/applogs/  
heap_trace.txt -XX:-HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/data/applogs/HeapDumpOnOutOfMemoryError
```

频繁FullGC可能的原因：

- 内存泄漏
- 大对象： Local Cache、无界队列、业务大对象etc.



► Jstat

► 轻量级jvm监控工具，包含了对堆内存使用情况和GC状况的监控

► jstat -gc <vmid> : 使用量

SOC	S1C	S0U	S1U	EC	EU	OC	OU	PC	PU	YGC	YGCT	FGC	FGCT	GCT
43648.0	43648.0	3813.0	0.0	961280.0	887088.1	1572864.0	532980.5	131072.0	66023.6	18216	213.160	4	0.213	213.374
43648.0	43648.0	3813.0	0.0	961280.0	923021.3	1572864.0	532980.5	131072.0	66023.6	18216	213.160	4	0.213	213.374
43648.0	43648.0	3813.0	0.0	961280.0	954958.5	1572864.0	532980.5	131072.0	66023.6	18216	213.160	4	0.213	213.374
43648.0	43648.0	0.0	2300.5	961280.0	43113.7	1572864.0	532997.6	131072.0	66023.6	18217	213.169	4	0.213	213.383
43648.0	43648.0	0.0	2300.5	961280.0	87764.8	1572864.0	532997.6	131072.0	66023.6	18217	213.169	4	0.213	213.383
43648.0	43648.0	0.0	2300.5	961280.0	130117.4	1572864.0	532997.6	131072.0	66023.6	18217	213.169	4	0.213	213.383

► jstat -gcutil <vmid> : 百分比

S0	S1	E	O	P	YGC	YGCT	FGC	FGCT	GCT
0.00	11.23	74.80	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	78.02	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	81.68	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	85.10	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	90.40	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	95.11	33.19	50.37	18139	212.249	4	0.213	212.463
0.00	11.23	98.80	33.19	50.37	18139	212.249	4	0.213	212.463
13.06	0.00	3.38	33.19	50.37	18140	212.267	4	0.213	212.480
13.06	0.00	7.30	33.19	50.37	18140	212.267	4	0.213	212.480
13.06	0.00	11.88	33.19	50.37	18140	212.267	4	0.213	212.480
13.06	0.00	15.83	33.19	50.37	18140	212.267	4	0.213	212.480

JVM工具



美团点评

► Jmap

► jvm中内存对象分布情况

► jmap -histo <vmid>

► [- 数组

► L - 对象引用

num	#instances	#bytes	class name
1:	12416225	397319200	java.util.HashMap\$Entry
2:	1727163	187337336	[B
3:	2036094	163695240	[C
4:	150477	62055864	[I
5:	3175610	50809760	java.lang.Integer
6:	463380	45272280	[Ljava.lang.Object;
7:	1360019	32640456	java.lang.String
8:	123694	32295672	[Ljava.util.HashMap\$Entry;
9:	633291	25331640	java.math.BigDecimal
10:	400820	19239360	sun.nio.cs.UTF_8\$Encoder
11:	117252	17765600	<constMethodKlass>
12:	117252	15021200	<methodKlass>
13:	541869	13004856	java.util.ArrayList
14:	10783	12745800	<constantPoolKlass>
15:	182706	8769888	java.util.HashMap
16:	10778	7854824	<instanceKlassKlass>
17:	8697	6986592	<constantPoolCacheKlass>
18:	53331	4693128	com.dianping.cat.message.spi.internal.DefaultMessageTree
19:	48674	3893920	com.caucho.hessian.io.Hessian2Input
20:	58947	2829456	com.dianping.cat.message.internal.DefaultEvent
21:	5380	2716456	<methodDataKlass>
22:	76435	2445920	java.util.ArrayList\$Itr
23:	30748	2213856	com.dianping.cat.message.internal.DefaultTransaction
24:	52615	2104600	java.util.HashMap\$EntryIterator
25:	63750	2040000	java.lang.ThreadLocal\$ThreadLocalMap\$Entry
26:	47728	2016168	[Ljava.lang.String;
27:	20089	1964928	[Z
28:	40711	1954128	com.caucho.hessian.io.SerializerFactory
29:	46250	1850000	java.util.LinkedHashMap\$Entry
30:	76896	1845504	java.lang.StringBuilder
31:	70656	1695744	io.netty.buffer.PoolThreadCache\$MemoryRegionCache\$Entry
32:	41912	1676480	java.util.HashMap\$KeyIterator
33:	25491	1631424	java.nio.DirectByteBuffer
34:	48849	1563168	java.util.concurrent.locks.AbstractQueuedSynchronizer\$Node
35:	13249	1483888	java.util.GregorianCalendar
36:	36328	1453120	redis.clients.jedis.Response

JVM工具



▶ Jstack:

1. 定位java进程，找出cpu使用率最高的PID
2. 定位线程 (**top -p <PID> -H**)
3. 线程号转化为16进制 (**printf "%x\n" tid**)
4. **jstack [-l][F] pid | grep tid**
5. 结合线程堆栈信息分析定位问题代码

▶ 需关注的线程状态有：

- 死锁， **Deadlock (重点关注)**
- 执行中， **Runnable**
- 等待资源 **Waiting on condition (重点)**
- 等待获取监视器， **Waiting on monitor entry (重点关注)**
- 暂停， **Suspended**
- 对象等待中， **Object.wait()**
或 **TIMED_WAITING**
- 阻塞， **Blocked (重点关注)**
- 停止， **Parked**

```
"Swallow-Helper-1" prio=10 tid=0x00002b15a4763000 nid=0x2c9 waiting on condition [0x00002b15bb8ab000]
    java.lang.Thread.State: TIMED_WAITING (sleeping)
        at java.lang.Thread.sleep(Native Method)
        at java.lang.Thread.sleep(Thread.java:340)
        at java.util.concurrent.TimeUnit.sleep(TimeUnit.java:356)
        at com.dianping.swallow.consumer.internal.task.LongTask$1.run(LongTask.java:71)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
        at java.lang.Thread.run(Thread.java:745)

"slaveConsumer-1" prio=10 tid=0x00002b15a47c2000 nid=0x2c8 waiting for monitor entry [0x00002b15bb829000]
    java.lang.Thread.State: BLOCKED (on object monitor)
        at com.dianping.swallow.consumer.internal.ConsumerThread.run(ConsumerThread.java:88)
        - waiting to lock <0x0000000788908d70> (a io.netty.bootstrap.Bootstrap)
"masterConsumerThread" prio=10 tid=0x00002b15a47c1000 nid=0x2c7 in Object.wait() [0x00002b15bb829000]
    java.lang.Thread.State: WAITING (on object monitor)
        at java.lang.Object.wait(Native Method)
        - waiting on <0x0000000788a23388> (a io.netty.channel.AbstractChannel$CloseFuture)
        at java.lang.Object.wait(Object.java:503)
        at io.netty.util.concurrent.PromiseFuture$DefaultPromise.await(PromiseFuture.java:254)
        - locked <0x0000000788a23388> (a io.netty.channel.AbstractChannel$CloseFuture)
        at io.netty.channel.DefaultChannelPromise.await(DefaultChannelPromise.java:129)
        at io.netty.channel.DefaultChannelPromise.await(DefaultChannelPromise.java:28)
        at com.dianping.swallow.consumer.internal.ConsumerThread.run(ConsumerThread.java:104)
        - locked <0x0000000788908d70> (a io.netty.bootstrap.Bootstrap)
```

The diagram shows three separate Java thread stacks from a jstack dump. Each stack has a call tree starting with a method name and its line number. Three specific states are highlighted with blue rounded rectangles and arrows pointing to them:

- Timed_waiting**: Points to the first stack, which is in a `TIMED_WAITING` state due to a `sleep` operation.
- Blocked**: Points to the second stack, which is in a `BLOCKED` state because it is waiting to lock a monitor object (`<0x0000000788908d70>`).
- Waiting**: Points to the third stack, which is in a `WAITING` state due to a `Object.wait()` call.

JVM工具



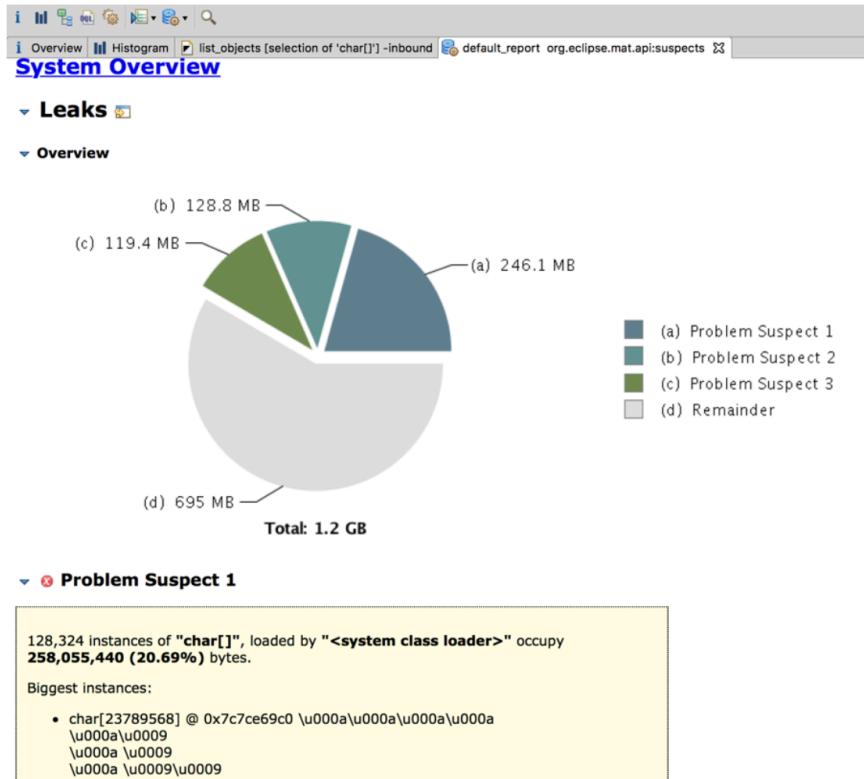
美团点评

› Jmap

- 复杂的问题需要Dump内存进行分析
- jmap –dump:format=b, file=/tmp/heap.dump <vmid>

› MAT:

- Histogram
 - 内存中对象的个数以及大小。
- Dominator Tree
 - 有哪些线程，以及线程下的对象占用的空间。
- Top consumers
 - 通过图形列出最大的object。
- Leak
 - Suspects通过MA自动分析泄漏的原因
- 注：需勾选Keep unreachable objects



网络工具



美团点评

▶ 网络连接状态：

▶ netstat -tnp

Active Internet connections (w/o servers)						
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/PPID
tcp	0	0	10.1.4.171:80	10.69.37.7:53240	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.193.2:20565	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.197.2:443	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.197.2:443	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.16.10:443	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.37.8:55664	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.37.10:43478	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.164.137:26962	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.7.11:8433	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.165.20:56518	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.37.7:20001	ESTABLISHED	-
tcp	0	0	10.1.4.171:58422	10.1.124.92:51867	ESTABLISHED	-
tcp	0	0	10.1.4.171:80	10.69.197.2:24869	ESTABLISHED	-

接收队列

发送队列

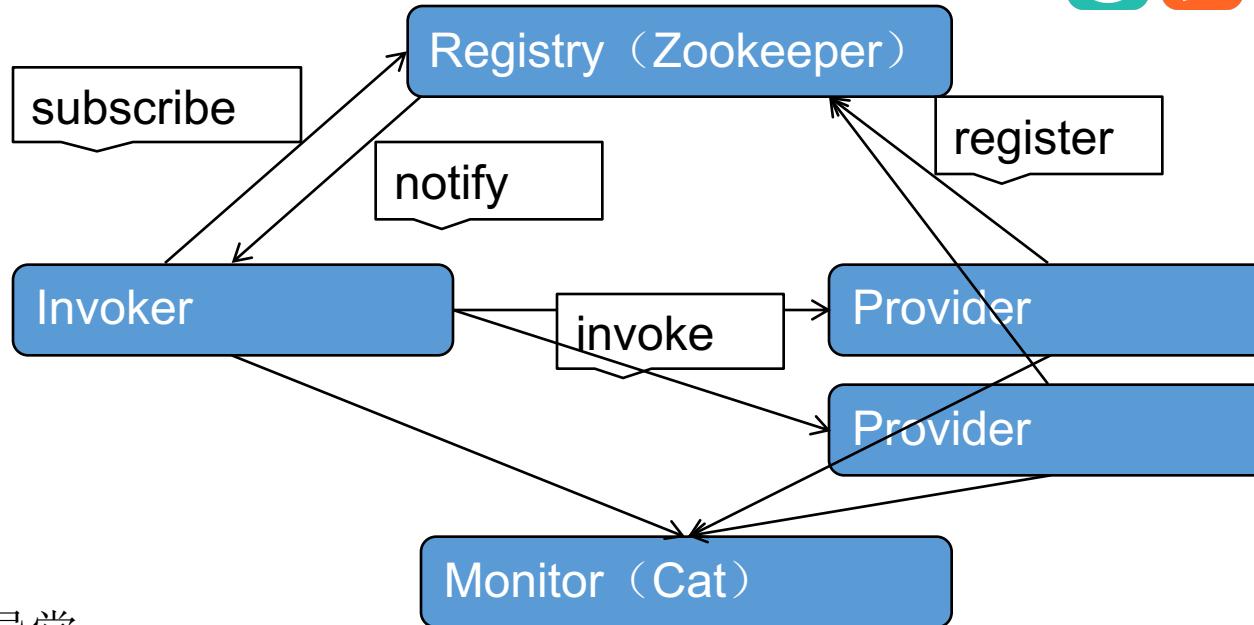
SLB

▶ 网络重传统计：

▶ netstat -s | grep "retransmt"

```
[yunshuang.tao@overseas-mobile-api-web01 ~]$  
[yunshuang.tao@overseas-mobile-api-web01 ~]$ netstat -s | grep "retran"  
    76309 segments retransmited  
      692 fast retransmits  
     158 forward retransmits  
   1187 retransmits in slow start  
     6 sack retransmits failed  
[yunshuang.tao@overseas-mobile-api-web01 ~]$
```

中间件与
运维工具



Pigeon关键异常

- com.dianping.pigeon.remoting.invoker.exception.ServiceUnavailableException
- com.dianping.pigeon.remoting.invoker.exception.RemoteInvocationException
- com.dianping.pigeon.remoting.provider.exception.RequestAbortedException
- com.dianping.pigeon.remoting.provider.exception.ProcessTimeoutException
- com.dianping.pigeon.remoting.common.exception.RejectedException

Pigeon隔离之快慢线程池

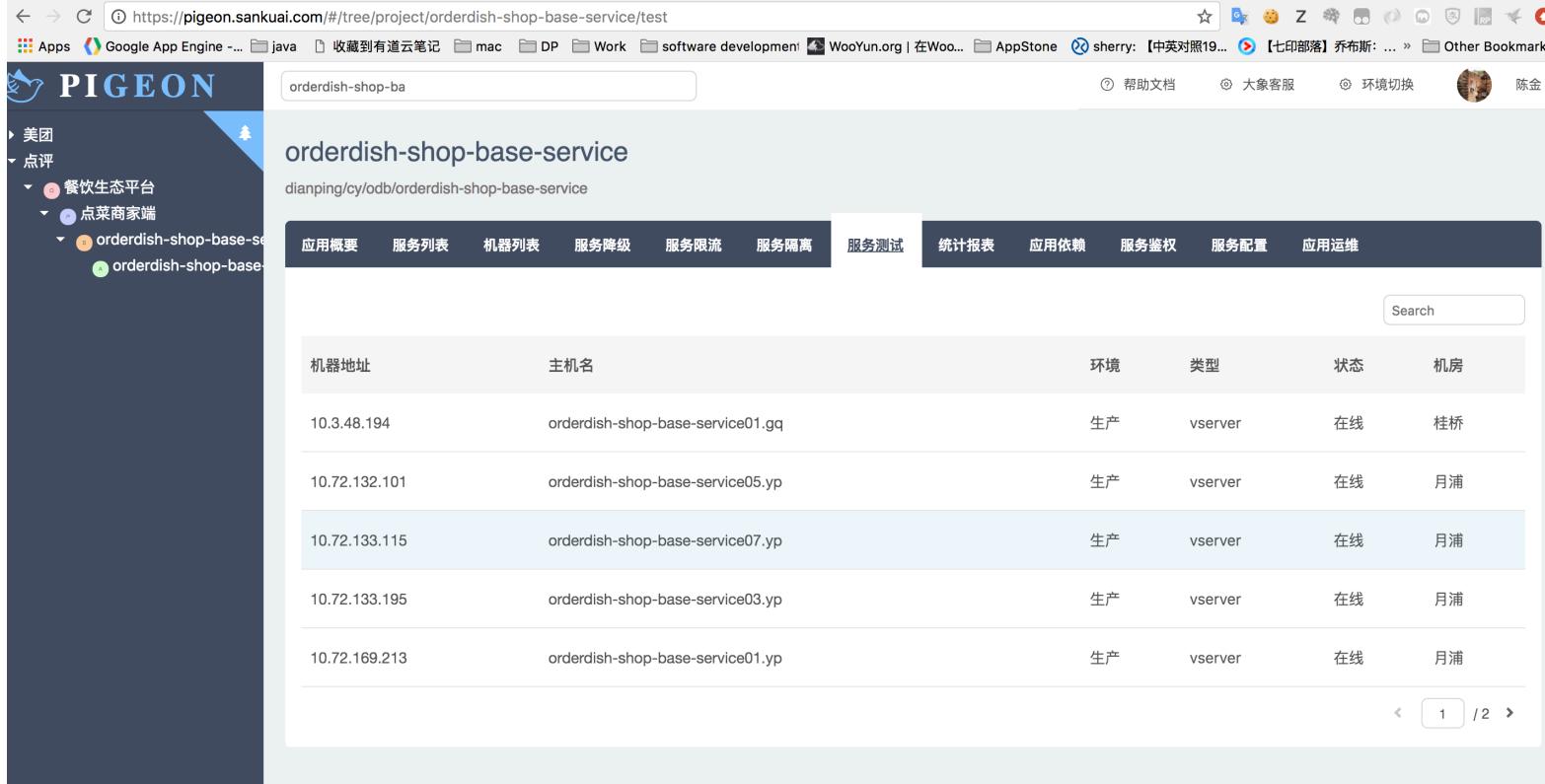
- RequestThreadPoolProcessor: *sharedRequestProcessThreadPool*、*slowRequestProcessThreadPool*
- *pigeon.provider.pool.slow.enable*, default true;
- *pigeon.provider.pool.slow.coresize*, default 30;
- *pigeon.provider.pool.slow.maxsize*, default 200;
- *pigeon.provider.pool.slow.queuesize*, default 500;
- *pigeon.provider.pool.coresize*, default 60;
- *pigeon.provider.pool.maxsize*, default 500;
- *pigeon.provider.pool.queuesize*, default 1000.

自定义配置:

- *appName. pigeon.provider.pool.slow.enable*,
- *appName. pigeon.provider.pool.slow.coresize*
- *appName. pigeon.provider.pool.slow.maxsize*,
- *appName. pigeon.provider.pool.coresize*, etc.

重启生效

服务治理平台



The screenshot shows the Pigeon service governance platform interface. The left sidebar navigation includes '美团' (Meituan), '点评' (Dianping), '餐饮生态平台' (Food & Beverage Ecosystem Platform), '点菜商家端' (Order菜 Merchant End), and two entries under 'orderdish-shop-base-service'. The main content area displays the 'orderdish-shop-base-service' service details, including its URL (dianping/cy/odb/orderdish-shop-base-service). A navigation bar at the top of the main content area includes tabs for '应用概要' (Application Overview), '服务列表' (Service List), '机器列表' (Machine List), '服务降级' (Service Degradation), '服务限流' (Service Flow Control), '服务隔离' (Service Isolation), '服务测试' (Service Testing) (which is selected), '统计报表' (Statistics Report), '应用依赖' (Application Dependency), '服务鉴权' (Service Authentication), '服务配置' (Service Configuration), and '应用运维' (Application Operations & Maintenance). Below the navigation bar is a search bar labeled 'Search'. The main content area contains a table with columns: '机器地址' (Machine Address), '主机名' (Host Name), '环境' (Environment), '类型' (Type), '状态' (Status), and '机房' (Data Center). The table lists five instances of the service:

机器地址	主机名	环境	类型	状态	机房
10.3.48.194	orderdish-shop-base-service01.gq	生产	vserver	在线	桂桥
10.72.132.101	orderdish-shop-base-service05.yp	生产	vserver	在线	月浦
10.72.133.115	orderdish-shop-base-service07.yp	生产	vserver	在线	月浦
10.72.133.195	orderdish-shop-base-service03.yp	生产	vserver	在线	月浦
10.72.169.213	orderdish-shop-base-service01.yp	生产	vserver	在线	月浦

Pagination controls at the bottom right indicate page 1 of 2.

缓存



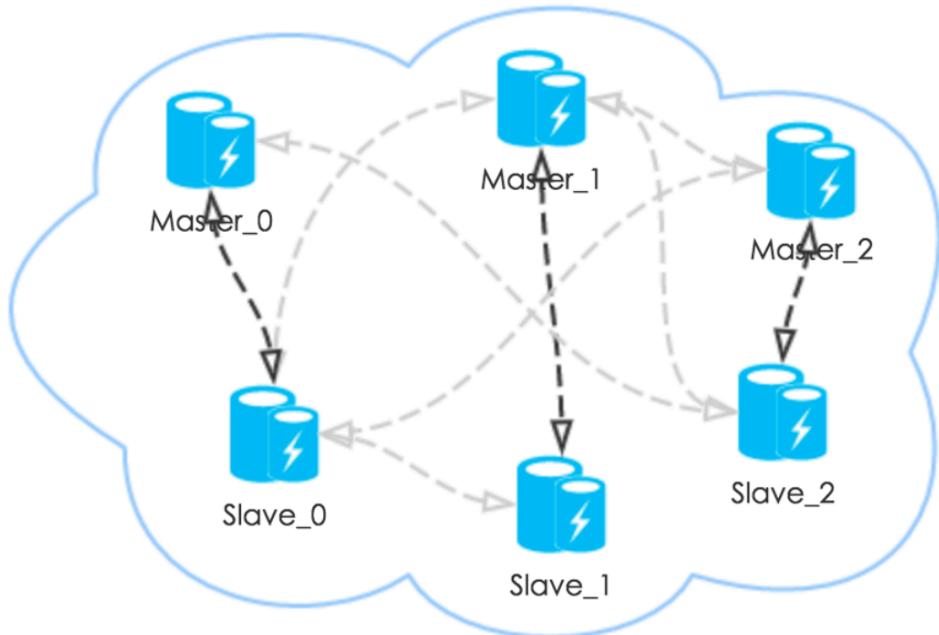
lion动态配置客户端参数

```
appName.squirrel.redis.conn.params  
key和value按需选填, eg:  
poolMaxTotal=30&poolMaxIdle=10  
&asyncCoreSize=20&asyncMaxSize=20  
&asyncQueueSize=1500&readTimeout=10  
0&routerType=master-slave
```

详细请参见: [Squirrel客户端参数详解](#)

容量上限

单节点约为7w+



缓存



← → C ⓘ rms.sankuai.com/cluster/info?cluster=redis-orderdish-shop

Apps Google App Engine ... Java 收藏到有道云笔记 mac DP Work software development! WooYun.org | 在Woo... AppStone sherry: 【中英对照19...】七印部落】乔布斯: ... » Other Bookma

Redis 监控平台

redis监控平台 > redis-orderdish-shop

我的集群

redis-movie

redis-orderdish-shop

redis-orderdish-shop

总览 慢查询 大key 集群治理 CAT实时报错分析

集群实时总QPS

数据量变化趋势 (GB)

节点状态

QPS 倾斜: 良好 内存倾斜: 良好

QPS 内存使用率 节点网卡流量 10分钟CAT报错

桂花 桂桥 月浦 光环

主节点

集群容量

总体容量 1.23 / 9 G 14 %

最大节点 0.14 / 1 G 14 %

最小节点 0.14 / 1 G 14 %

数据库

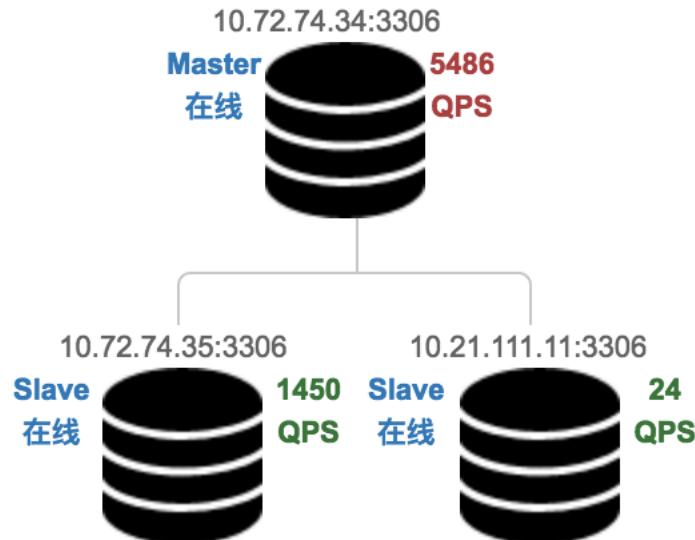


美团点评

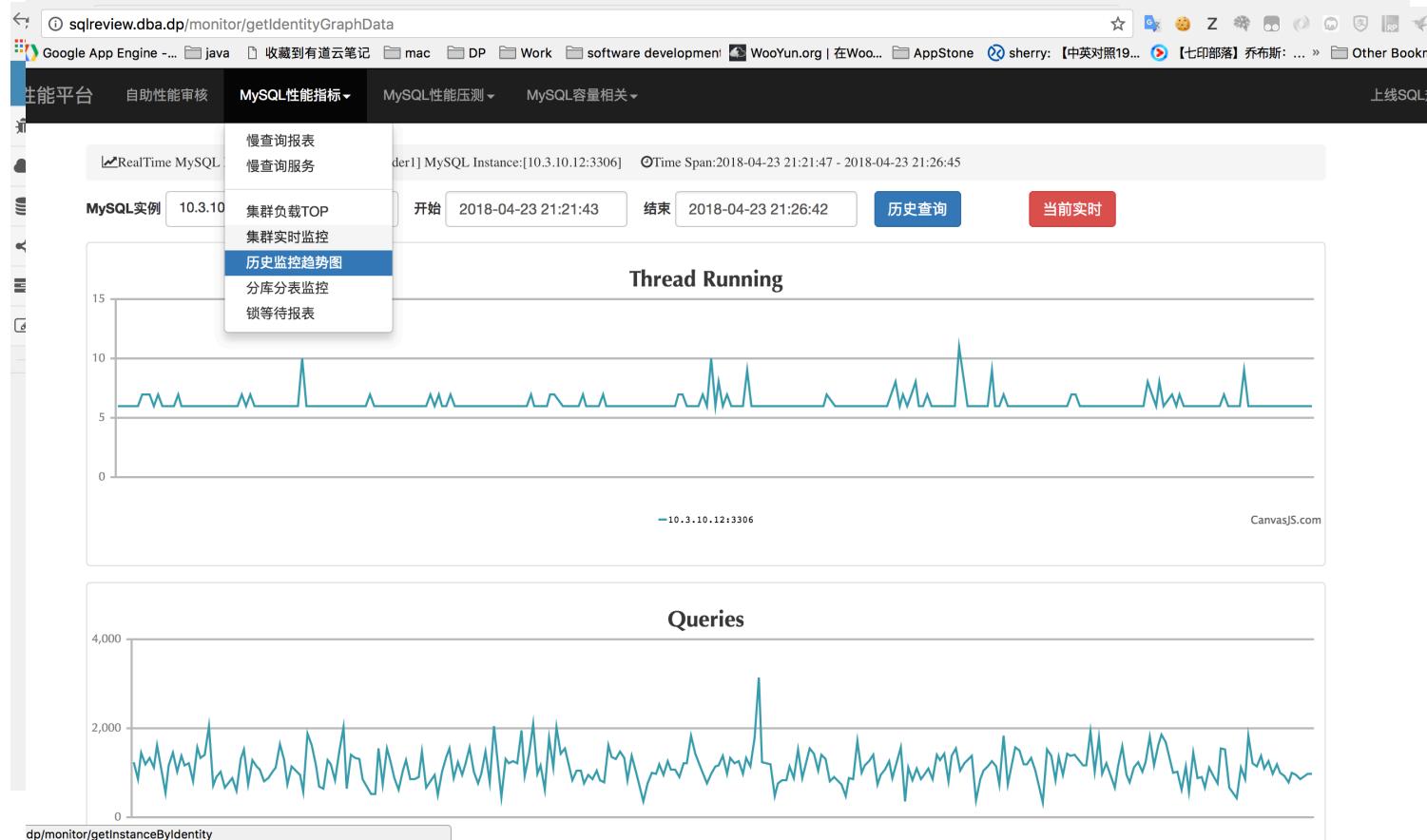
Router Type

```
@Deprecated  
ROUND_ROBIN("round-robin"),  
MASTER_SLAVE("master-slave"),  
@Deprecated  
LOAD_BALANCE("load-balance"),  
SLAVE_ONLY("slave-only"),  
@Deprecated  
FAIL_OVER("fail-over"),  
MASTER_ONLY("master-only");
```

主从延迟



数据库

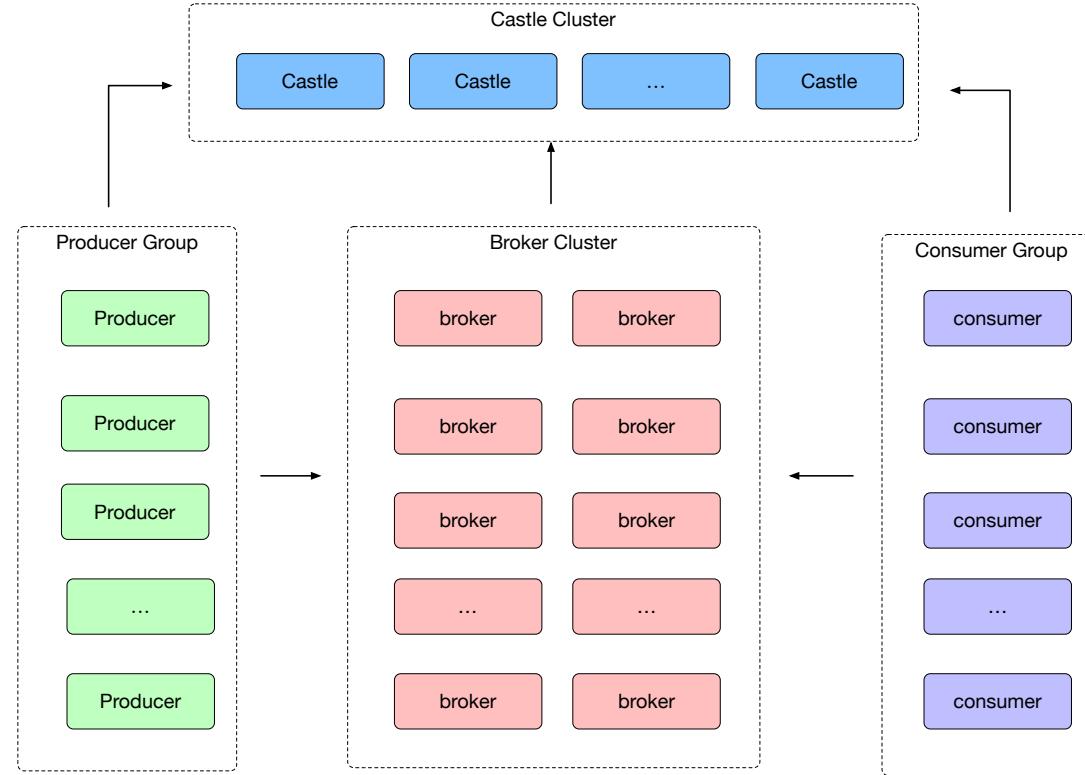


mafka partition机制

一个topic中的一个partition只能被一个消费组中的一个consumer消费。在mafka中，一个consumer为一个消费线程。

消费者数 * 消费线程数 <= partition数

errorCode:NO_PARTITION_ASSIGN





Secure | <https://mq.sankuai.com/subscribe/94555/selectedTime/>

Apps Google App Engine ... java 收藏到有道云笔记 mac DP Work software development! WooYun.org 在Woo... AppStone sherry: 【中英对照19... 【七印部落】乔布斯: ... Other Bookmarks

MQ管理平台

主题管理

主题申请

主题列表

审批列表

消费组管理

消费组列表

消费组申请

消费组审批列表

Appkey管理

Appkey列表

报警管理

报警列表

新增报警

报警日志

消费组 - orderdish_bapp_service

消费组名称	主题	消费者Appkey	状态	集群状态	描述
orderdish_bapp_service	orderdish_order_accept_notify	orderdish-bapp-service	运行中	dx-mafka2-daocan 大兴 yp-mafka2-daocan 月浦	

操作

删除消费组 修改配置 消费回溯/前进 添加消费组报警 停用消费组 按时间查看监控 开启死信

详情

监控 报警项列表 配置报警人 业务治理 demo示例

时间段: [1小时] [2小时] [4小时] [6小时] [8小时] [12小时] [24小时] [48小时] [falcon查询说明文档](#)

积压数目

-○- dx-mafka2-daocan当前积压个数 -○- yp-mafka2-daocan当前积压个数

The chart displays the current backlog count for two Kafka clusters: dx-mafka2-daocan (blue line) and yp-mafka2-daocan (orange line). The Y-axis represents the backlog count, ranging from 0.8 to 1. The X-axis represents time intervals. Both clusters show a backlog count of approximately 0.8 across all time intervals.

时间段	dx-mafka2-daocan (蓝色)	yp-mafka2-daocan (橙色)
[1小时]	~0.8	~0.8
[2小时]	~0.8	~0.8
[4小时]	~0.8	~0.8
[6小时]	~0.8	~0.8
[8小时]	~0.8	~0.8
[12小时]	~0.8	~0.8
[24小时]	~0.8	~0.8
[48小时]	~0.8	~0.8

TTT流程与服务树



美团点评

常用流程

- 应用重启
- JVM dump
- 应用扩容
- 静默期发布
- 清理CDN缓存

美团点评运维 [服务树](#) 流程中心 预算系统 API 文档 使用手册 反馈入口 搜索... SRE 负责人: yangfan32 设置 陈金 退出

dianping

- cy
 - ecard
 - caterdb
 - dcrai
 - dj
 - ft
 - idp
 - odb
 - oodc
 - odd
 - odm
 - qa
 - sam
 - ssp
- life
- meituan

节点信息 机器信息 域名管理 监控管理 配置管理 权限管理 资源利用率 VS 管理 调用链 corp=meituan&owt=redis&pdl=cy&srv=orderdishshop

主机列表 监控指标 物理拓扑 刷新数据 所有环境 prod staging beta test dev Show 60 entries, search: 导出主机列表

主机	IP	CPU	MEM	NIC	宿主机	CPU 型号	集群	操作
set-gh-redis-orderdish-shop01	10.22.53.90	4	2	双万兆	gh-cloud-docker-hulk343	E5-2650v4@2.20GHz		<button>登录</button>
set-gh-redis-orderdish-shop02	10.22.52.97	4	2	双万兆	gh-cloud-docker-hulk376	E5-2650v4@2.20GHz		<button>登录</button>
set-gh-redis-orderdish-shop03	10.20.97.124	4	2	双万兆	gh-cloud-docker-hulk107	E5-2630v4@2.20GHz		<button>登录</button>

服务树:

- [TTT](#)
- [OPS](#)

T.T.T 应用树 发起流程 流程纪录 资源管理 HTTP 探测 MGW 故障修复 报警管理 搜索 应用/IP/主机名 文档 管理 陈金

应用树 > 餐饮生态平台 > 餐饮生态DB与存储 > cater-db-mysql ❤

节点: corp=dianping&owt=cy&pdl=caterdb&srv=cater-db-mysql&cluster=prod
负责人: guofeng08,hancheng02,fanfugui,wangjian57
运维: guofeng08
监控: cat | falcon

健康度 100 资源管理 应用详情 流程记录 发布记录 SLB 规则 监控管理 弹性伸缩 变更记录 快捷操作

● 全部 ○ 生产环境 ○ beta & ppe

生产 5	IP	泳道	类型	状态	CPU	内存	机房	上线时间	web终端	更多
mysql-caterdb01.gh	10.21.111.11	default	物理机	在线DB	-	-	光环	-	<button>登录</button>	<button>详情</button>
mysql-caterdb01.yp	10.72.63.25	default	物理机	在线DB	-	-	月浦	-	<button>登录</button>	<button>详情</button>
mysql-dcdb01.yp	10.72.74.34	default	物理机	在线DB	-	-	月浦	-	<button>登录</button>	<button>详情</button>

03

原则与手段

处理原则

应对手段

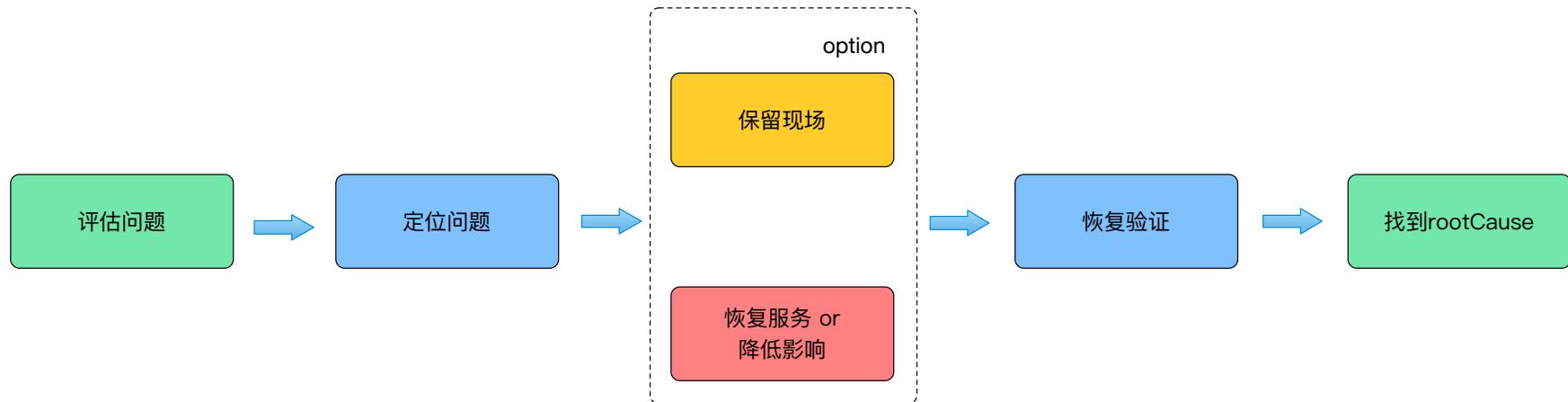
处理原则



第一时间减小损失

解决问题优先于找到Root Cause

如不具备条件，可不保留现场



应对手段



运维三板斧--重启

- <https://ttt.sankuai.com/#!/start/3>
- 应用负责人免审核，正确且非单点的应用负责人

运行时非突发流量故障

应对手段



运维三板斧--回滚

- 后端应用回滚：分组回滚 & 快速全量回滚
- 前端应用回滚：<https://123.sankuai.com/km/page/13238546>, 新旧6种组合
- 小程序回滚：版本管理-回滚至上一版本
- App回滚：撤回，只是降低影响，小版本 or hotfix才能解决问题
- 其他线上变更：数据备份，提前准备回滚脚本

有变更，先回滚
回滚！= 重新发布旧代码
上线方案需包含回滚方案

应对手段



运维三板斧--加机器

- 后端应用扩容: <https://ttt.sankuai.com/#!/start/1>
- Squirrel扩容: 扩分片 or 扩Slave节点
- 数据库扩容: 加从库
- MQ扩容: 扩consumer之前, 可能需要扩partition

运行时突发流量
找到资源瓶颈

应对手段



降级

- 非核心功能
 - ✓ 影响体验但不block交易流程
- 如何降级
 - ✓ 业务降级开关
 - ✓ 服务限流/服务降级
 - ✓ MQ Consumer黑名单
 - ✓ RDS可访问应用白名单
 - ✓ SQL黑名单
 - ✓ SRE操作SLB
 - ✓ 暂停非核心作业

运行时突发流量

04



案例分析



循环调用几乎压垮db

美团排队线上入口404

info-bom升级导致ecom不可用

对接点付上线导致ddt不可用

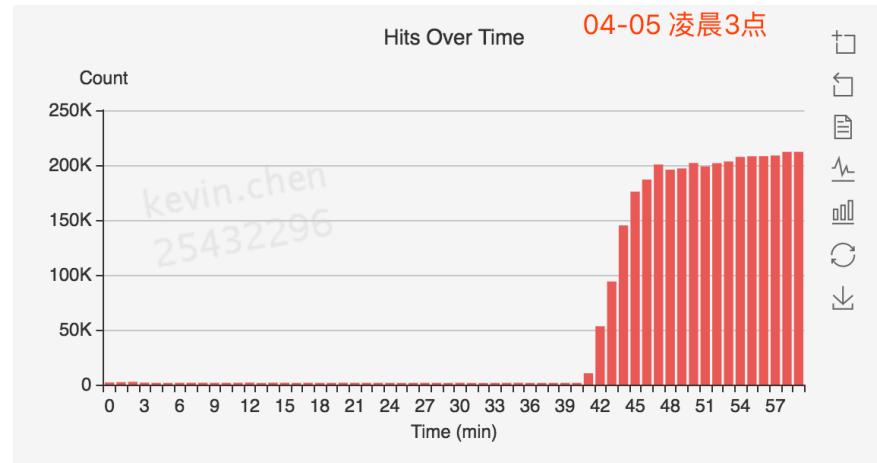
猫眼17年春节故障

循环调用几乎压垮db



问题概述

- 问题现象
orderdish-order-base-service访问量暴涨，QPS最高上升至6000+
- 起止时间
04.05 04:00 ~ 04.07 11:15
- 影响范围
无
- 造成损失
无



<https://wiki.sankuai.com/pages/viewpage.action?pageId=1467136905>

循环调用几乎压垮db



处理过程

- 发现问题

04-07 07:43 C端赵林反馈该接口从5号开始访问量暴增

- 评估影响

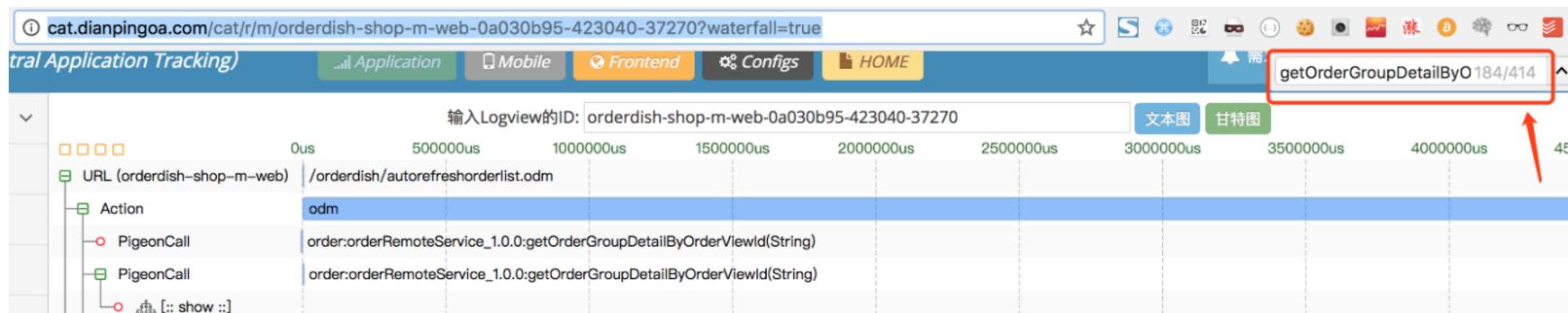
04-07 07:46 ~ 07:50 确认不影响业务

- 问题定位

04-07 07:55 确认shop-m-web无定时任务及最近的一次上线内容（3号上线升级jdk8）

04-07 08:10 cat分析发现3号4号接口调用量开始有毛刺，5号凌晨暴涨后不再下降

04-07 08:20 通过logview发现有接口的循环调用，一次action请求调用接口 400多次。



循环调用几乎压垮db



处理过程

- 解决问题

04-07 09:22 评估服务是否能够抗拒压力，发现压力已到DB，而且应用强制走主库

04-07 10:05 准备回滚应用，降级jdk版本，拉SRE、DBA入群。

04-07 10:30 回滚 & 降回tomcat&jdk版本没解决问题，明确若业务高峰解决不了则可把bapp autorefresh接口禁了。同时确认修改代码解决问题。

【 cater 基本信息 备份配置]

基本信息 » 查看该集群上部署的实例信息和所有的数据库信息

实例数:3	部门:餐饮生态平台
数据库数:24	负责人:guofeng08
状态: 在线	虚IP:10.72.74.34
修改时间:2018-04-04 10:24:56	MHA节点IP:10.1.101.50

55



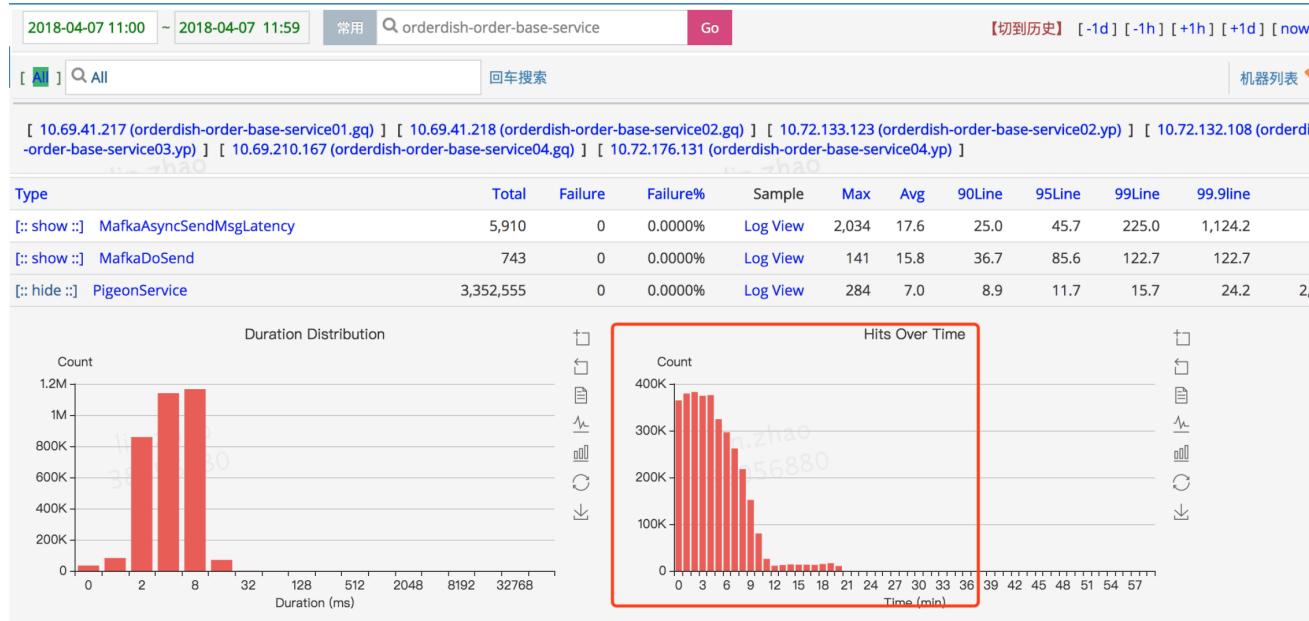
美团点评

循环调用几乎压垮db

处理过程

- 解决问题(续)

04-07 11:10 限制for循环次数代码上线，流量下降，业务不受影响。



循环调用几乎压垮db



美团点评

问题复盘

- 一次线上压测
orderdish-order-base-service简单查询接口，单机QPS 1000

- 暴露的问题

- ✓ 应用强制走主库
- ✓ 接口入参校验不完备
- ✓ 非按需获取数据
- ✓ 不受限制的for循环调用
- ✓ 业务代码写法复杂

```
List<String> orderViewIds = new ArrayList<>();
//查询自动状态更新订单
List<AutoPrintRecord> updatedAutoPrintRecords = printOrderRemoteService.queryAutoPrintRecordsByUpdateTime(startTime, endTime);
for (AutoPrintRecord autoPrintRecord : updatedAutoPrintRecords) {
    if(null != autoPrintRecord)
        orderViewIds.add(autoPrintRecord.getOrderViewId());
}
//查询手动状态更新订单
List<ManualPrintRecord> updatedManualPrintRecords = printOrderRemoteService.queryManualPrintRecordsByUpdateTime(startTime, endTime);
for (ManualPrintRecord manualPrintRecord : updatedManualPrintRecords) {
    if(null != manualPrintRecord)
        orderViewIds.add(manualPrintRecord.getOrderViewId());
}
if(CollectionUtils.isEmpty(orderViewIds)){
    return;
}

//将更新的订单加入返回结果
for (String orderViewId : orderViewIds) {
    HbtOrderGroupDetailBO orderDetail = hobbitOrderRemoteService.getOrderByOrderViewId(orderViewId);
    ...
}
```

美团排队404



问题概述

- 问题现象
原美团排队美团App POI页入口跳转404页面
- 起止时间
02-27 15:26 ~ 02-28 00:21
- 影响范围
sr.api.vip.sankuai.com域名无法访问
- 造成损失
28983个访问返回502，损失订单约300单。

<https://sr.api.vip.sankuai.com>/pages/viewpage.action?pageId=1459503822



美团点评

美团排队404

处理过程

- ## • 发现问题

02-27 21:08:00 产品反馈原美团排队h5页面(<http://ismart.meituan.com/queue/poi/xx>)404了

- 评估影响

排队订单线下入口占9成，订单影响小，但美团app POI页入口DAU高，404影响大

- 问题定位

02-27 21:12:00 RD开始定位，确认应用无上线，查看应用日志

02-27 21:26:00 将SRE拉进群一起定位，发现问题始于下午15点

美团排队404



美团点评

处理过程

- 问题定位(续)

02-27 21:12:00 RD开始定位，确认应用无上线，查看应用日志

02-27 21:26:00 将SRE、原维护美团排队的同学拉进群一起定位，杨凡确认问题始于15点

02-27 21:49:00 确认问题是poi入口页面是在后端接口报错时302到了404页面

02-27 21:50:00 决定重启应用，重启不生效

02-27 22:56:00 拉前端一起定位问题，但前端不熟悉该应用。自己开始撸代码。

```
module.exports = function(app) {
  //出现错误跳转到错误页
  app.use(function*(next) {
    if (!this.app.config.isOnline) {
      yield next;
      if ((404 <= this.status && 417 >= this.status)) {
        this.redirect('/error/404');
      }
      if (500 <= this.status){
        this.redirect('/error/500');
      }
    } else {
      try {
        yield next;
      } catch (e) {
        var msg = e.stack || e.toString();
        this.body = msg;
      }
    }
  });
};
```

```
function* QueueStatusByUserId(args) {
  var act = "queues";

  var uri = "http://" + this.config.get("host.paidui") + act;

  var param = {
    lng: "" + (args.lng || 0),
    lat: "" + (args.lat || 0),
    token: args.token,
    poiId: args.poiId,
    page: 0,
    size: 1,
    onlyToday: true
  };

  var requestObject = {
    uri: uri,
    qs: param,
    timeout: 1300,
    label: 'srq-' + act
  };
}
```

```
host: {
  // upstream timeout:
  // http://wiki.sankuai.com/pages/viewpage.action?pageId=1
  // CDN地址
  cdn: "meituan.net",
  // 主域名
  main: "meituan.com",
  // 网站域名
  document: "ismart.meituan.com",
  // 用户信息接口
  userInfo: "open-in.vip.sankuai.com",
  // 读取poi信息接口
  poi: "api.poi-in.sankuai.com",
  // 排队接口
  paidui: "sr.api.vip.sankuai.com/sup/capp/v1/",
  // 点菜接口
  diancaiv1: "sr.cms.sankuai.com/srcms/v1/",
  // 正餐查询订单详情
  order_info: "sr.cms.sankuai.com/srcms/v2/",
  // 微信排队接口
  wechat_kao: "sr.api.vip.sankuai.com/sup/wechat/queue/",
  // 门店分析
  analytics: "sr.crm.sankuai.com/sup/analysis/"
},
```

美团排队404



美团点评

处理过程

- 问题定位(续)

02-27 23:51:00~次日00:21 SRE徐方华加入排查，发现sr.api.vip.sankuai.com对应的后端不存在了。但RD同学确认直接ip访问机器是ok的。SRE杨凡确认配置升级删了nginx upstream.

```
[root@dx-nginx-daodian01 nginx]# grep '/sup/capp/v1/queues' /var/sankuai/wwwlogs/sr.e.sankuai.com-access_log.nginx.192.27.1.1 awk '{if($11==200) print $0}' |more
10.32.99.151 - [27/Feb/2018:15:37:37 +0800] 0.000 36352667966 "GET /sup/capp/v1/queues/150710910683923330 HTTP/1.1 502 269 "-" "-" sr.api.vip.sankuai.com - http - 30413 - - 1
10.32.99.151 - [27/Feb/2018:15:37:41 +0800] 0.001 36352725038 "GET /sup/capp/v1/queues?lng=111.28462082044183&lat=39.17510651068&token=fPK_0EVcekmc55B25zE0Y013wAAAAAAwUAABtZjmyuN8t366z14bWwZHPrSjXQ0mEpD5D7zDWH_laP4XpN5Q5r10DJlzZZbeg&poiId=40256865&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 30436 - - 1
10.32.99.151 - [27/Feb/2018:15:39:17 +0800] 0.000 36354221908 "GET /sup/capp/v1/queues?lng=113.322174&lat=23.141158&token=&poiId=2580887&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-"
"-" sr.api.vip.sankuai.com - http - 30876 - - 1
10.32.99.151 - [27/Feb/2018:15:39:12 +0800] 0.000 36354141128 "GET /sup/capp/v1/queues?lng=102.19952&lat=25.06701&token=JFokalDMjFA7Q27aaamuPMUTKeowAAAAAmgQAAI4h7001S04fLl4dVPD1qjbLooSR3iyGW
r91sSX4KdIG-WJ4E76BuEvTinxvQXeQ&poiId=6057877&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 30849 - - 1
10.32.99.151 - [27/Feb/2018:15:40:19 +0800] 0.000 36355167465 "GET /sup/capp/v1/queues?lng=113.555439&lat=22.286392&token=-RUUbEJGBjBzWdTc1kFmbxB0NHAAAAARQUAHabRDzmaUzMj0xWLlWnCERSNFYh6wNG7
5Me2N0J12837f00YCu9wnWxXZ0Wr80ow&poiId=24274238&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31186 - - 1
10.32.99.151 - [27/Feb/2018:15:40:44 +0800] 0.000 36355560248 "GET /sup/capp/v1/queues?lng=106.427713&lat=29.568603&token=0kNhNu0C6JHA-M8_VoYMoSGSA1gAAAAAAwUAACZn9de4rC6NQI5p92q6e66ASFvGDR0Y96
7YrwSMcXbQ8-7Zi85IVeZ2mkNdw&poiId=65836251&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31302 - - 1
10.32.99.151 - [27/Feb/2018:15:40:45 +0800] 0.000 36355576581 "GET /sup/capp/v1/queues?lng=113.517610&lat=22.260802&token=NKYtFqStLZYyRSvg1q4b3U3YRAAAAawUAAAA6oi1tv9lvkteYpGuocfS6NIqIjQBpxA
wfx-pEkqd4d3rBxLz9A6h6URxkFCGNQ0&poiId=61811110&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31311 - - 1
10.32.99.151 - [27/Feb/2018:15:40:46 +0800] 0.000 36355594323 "GET /sup/capp/v1/queues?lng=126.560698&lat=45.876228&token=SxhNEjR8IPp4Cc0T-8yZLFC4JAAAAAAwUAUAHTA4_BLLhV4t15Vp61GxguGKvaqdISqk1
CgCNQ-bbrIVdrBDL7p5mzAyembdqNsA&poiId=66740253&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31320 - - 1
10.32.99.151 - [27/Feb/2018:15:41:16 +0800] 0.000 36356048991 "GET /sup/capp/v1/queues?lng=120.544002&lat=31.38281&token=A1aL-VFxsn2FwBy86qZqT0lHBpUAAAAAAwUAAMXTsSvCnMpCfth_iK-KqX44RUQM8aQH-t6
jzf0ze71qNtfuG77ead6KH7l1lgT6DQ&poiId=90916228&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31449 - - 1
10.32.99.151 - [27/Feb/2018:15:41:45 +0800] 0.000 36356498628 "GET /sup/capp/v1/queues?lng=117.490518&lat=30.66855&token=e-G6ERA97jMNWDW5QLeK9-0JriQAAAAAAwUAAC0I01Ec_avegw5Na-1h72hsbFnVqbikZqE
fjnFHAl2QHf1k4LXet6tyX118FPV&poiId=40341528&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31573 - - 1
10.32.99.151 - [27/Feb/2018:15:42:18 +0800] 0.000 36357001669 "GET /sup/capp/v1/queues?lng=114.876207&lat=40.802112&token=W9czdyilB_zx-rXJn9QZ1X-h4IAAAAawUAAM9k4zOM4QKeRyn_6rHnMk7d_y1E-eeyg
GdqrQujRxRxf2rIiTad-NFFxiu4G6IF8Q&poiId=45213658&page=0&size=1&onlyToday=true HTTP/1.1" 502 269 "-" "-" sr.api.vip.sankuai.com - http - 31715 - - 1
```

- 解决问题

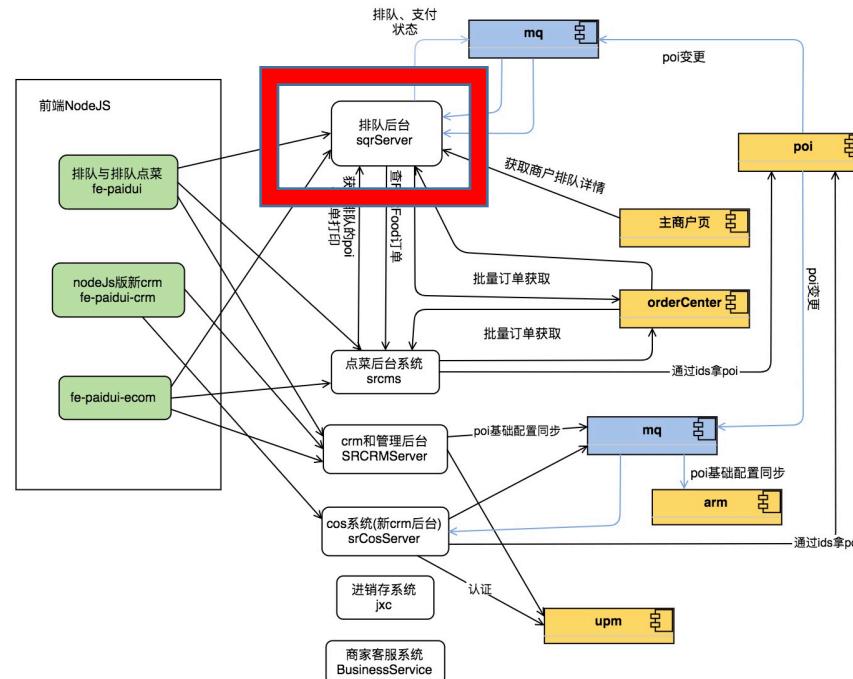
02-28 00:21 SRE杨凡恢复sr_api upstream配置，服务恢复

美团排队404



问题复盘

- 暴露的问题
 - ✓ 线上变更不周知
 - ✓ 美团侧技术栈不熟悉
 - ✓ 系统关键链路不熟
 - ✓ 监控缺失



Info-Bom升级导致ecom不可用



问题概述

- 问题现象

ecom菜品管理、门店管理页和旧订单页提示“服务不可用，稍后重试”

- 起止时间

01-12 23:00 ~ 01-13 12:03

- 影响范围

01-13 09:14 ~ 01-13 12:03 商家后台菜品管理、门店管理页和旧订单页不可用

- 造成损失

8172 个请求失败、92 家门店排队登录失败。

Info-Bom升级导致ecom不可用



处理过程

- 发现问题

01-13 09:14 orderdish-shop-web 出现 RemoteException 告警

01-13 10:48 值班群反馈商家登录开店宝提示“服务不可用，稍后重试”

- 评估影响

涉及商家菜品管理、门店管理以及部分订单管理，需快速恢复

- 问题定位

1月 13 日 11:13 B 端微信群开始跟进处理，值班同学判定是RPC的问题

1月 13 日 11:38 重启 orderdish-shop-web 应用，发现没有效果

1月 13 日 11:53 定位到是 orderdish-shop-facade-service 问题，重启 orderdish-shop-facade-service 第一组服务

1月 13 日 12:03 orderdish-shop-facade-service 重启完毕，线上恢复正常

Info-Bom升级导致ecom不可用



处理过程

- 问题定位(续)

1月13日 12:13 拉Cat/Pigeon相关同学拉进群定位

1月13日 14:20 定位到 AbstractMethodError 报错导致 Pigeon 未能 complete cat transaction打点，transaction 嵌套最终导致stackOverFlow，进而导致服务不可用，同时嵌套消息体导致cat client上报延迟，告警不发。AbstractMethodError则是由于 inf-bom 升级，info-bom mtrace版本和强制指定版本冲突。

```
ERROR java.lang.AbstractMethodError: com.meituan.mtrace.config.sg.SampleConfigByMcc.getLocalSwimlane()Ljava/lang/String;
at com.meituan.mtrace.ServerTracer.recordAsync(ServerTracer.java:70)
at com.meituan.mtrace.ServerTracer.record(ServerTracer.java:36)
at com.meituan.mtrace.Tracer.serverRecv(Tracer.java:179)
at com.dianping.pigeon.monitor.mtrace.ProviderMtraceTransaction.<init>(ProviderMtraceTransaction.java:78)
at com.dianping.pigeon.monitor.mtrace.MtraceMonitor.createTransaction(MtraceMonitor.java:46)
at com.dianping.pigeon.monitor.CompositeMonitor.createTransaction(CompositeMonitor.java:67)
at com.dianping.pigeon.remoting.provider.process.filter.MonitorProcessFilter.invoke(MonitorProcessFilter.java:84)
at com.dianping.pigeon.remoting.provider.filter.MonitorProcessFilter.invoke(MonitorProcessFilter.java:40)
at com.dianping.pigeon.remoting.provider.ProcessHandlerFactory$1.handle(ProcessHandlerFactory.java:98)
at com.dianping.pigeon.remoting.provider.ProcessFilter.invoke(TraceFilter.java:39)
at com.dianping.pigeon.remoting.provider.filter.TraceFilter.invoke(TraceFilter.java:14)
at com.dianping.pigeon.remoting.provider.ProcessHandlerFactory$1.handle(ProcessHandlerFactory.java:98)
at com.dianping.pigeon.remoting.provider.ProcessThreadPoolProcessor$1.call(RequestThreadPoolProcessor.java:19)
at com.dianping.pigeon.remoting.provider.ProcessThreadPoolProcessor$1.call(RequestThreadPoolProcessor.java:18)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

- 问题解决

重启已恢复服务，但风险依然在

1月13日 14:29 RD修改代码，清理排查 inf-bom 冲突 jar 包，测试环境验证

1月13日 16:12 hotfix代码发布上线

rice-0a48b6bc-421060-254?domain=orderdish-shop-facade-service		
	T12:47:37.589	PigeonService
	T12:46:57.833	PigeonService
	T12:46:51.465	PigeonService
	T12:45:32.814	PigeonService
	T12:45:25.469	PigeonService
	T12:44:36.372	PigeonService
	T12:44:06.561	PigeonService
	T12:42:52.640	PigeonService
	T12:41:50.102	PigeonService
	T12:41:46.217	PigeonService
	T12:41:33.671	PigeonService
	T12:40:38.757	PigeonService
	T12:40:32.155	PigeonService
	T12:40:06.145	PigeonService
	T12:39:52.683	PigeonService
	T12:39:24.136	PigeonService

Info-Bom升级导致ecom不可用



问题复盘

- 暴露的问题
 - ✓ 上线观察点
 - ✓ 线上验证
 - ✓ 组件升级sop
 - ✓ Pigeon隐藏风险
 - ✓ 单一报警手段存在不work的风险
 - ✓ 定位问题&解决问题的能力

对接点付上线导致ddt不可用



问题概述

- 问题现象

达店通接单全部失败，云打印订单没有取餐号，

- 起止时间

01-03 11:14 ~ 01-03 18:50

- 影响范围

达店通渠道接单2小时不可用，云打印机等渠道订单7+小时不展示取餐号

- 造成损失

达店通接单失败数1969，失败单数占比6.5%

<https://wiki.sankuai.com/pages/viewpage.action?pageId=1276025938>

对接点付上线导致ddt不可用



处理过程

1月3日 10:16 轻餐hack上线barCode改造

1月3日 11:14 D端收到bd反馈达店通接单全部失败

1月3日 11:57 B端收到D端反馈并开始排查问题

1月3日 12:02 定位问题是达店通接单时，反查C端订单时拿不到barCode抛异常导致接单失败，原因是上午上线的barCode改造

1月3日 12:15 B、C、D端讨论后认为回滚hobbit-base、订单中心、接单中心等核心项目风险太大，决定由D端紧急修复上线

1月3日 12:28 D端完成修复并开始上线

1月3日 12:31 第一组机器上线完成，观察线上接单情况，发现部分已恢复

1月3日 12:40 完成剩余两组机器上线，达店通接单渠道全部恢复

对接点付上线导致ddt不可用



处理过程(续)

1月3日 18:06 收到产品反馈云打印订单没有取餐号

1月3日 18:08 确认同样是取餐码改造导致的问题

1月3日 18:18 跟金哥确认通过高峰期紧急上线修复该问题

1月3日 18:26 第一组机器上线完成，关闭其他机器流量，查询日志发现打印任务中已经有了取餐码

1月3日 18:40 所有机器上线完成，流量控制放开

1月3日 18:50 BD确认小票取餐号恢复正常

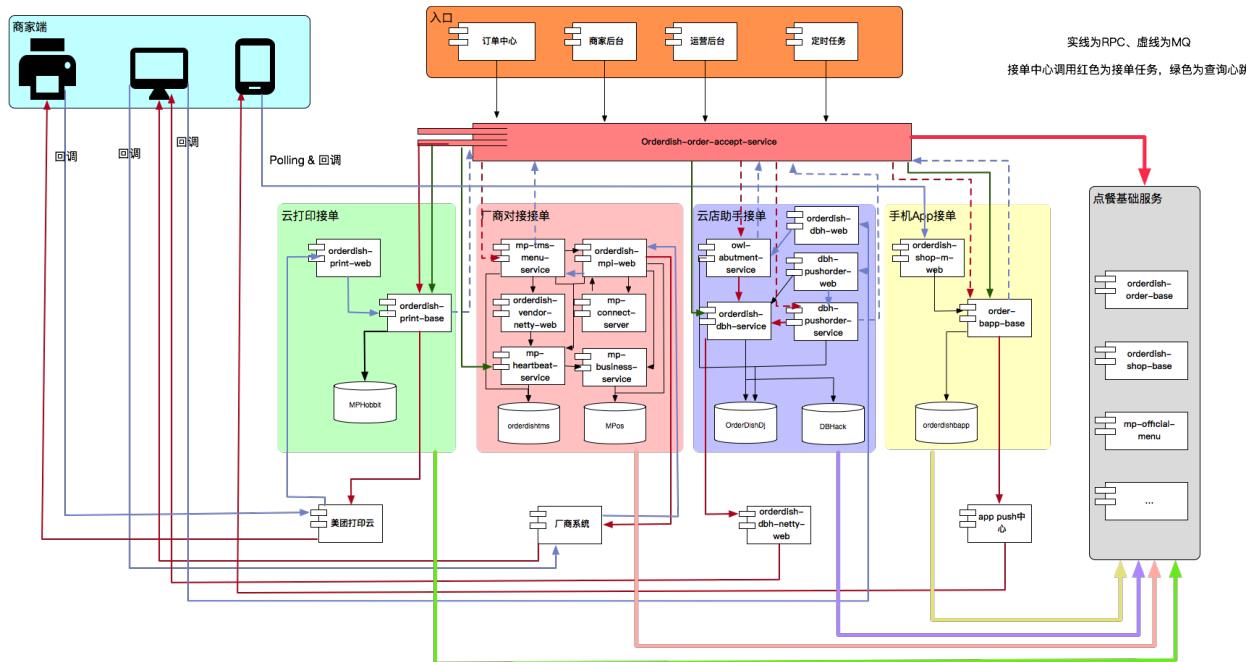
对接点付上线导致ddt不可用



美团点评

问题复盘

- 暴露的问题
 - ✓ 技术方案评审
 - ✓ 上线无回滚方案
 - ✓ 故障处理流程
 - ✓ 分渠道监控能力
 - ✓ 复杂的系统架构



17年猫眼春节故障



问题概述

- 问题现象

美团app电影频道、猫眼App影院列表、影院页影讯不能正常展示

- 起止时间

2017-01-28 10:40 ~ 18:14

- 影响范围

01-28 10:40 – 12:00、12:20 – 13:10、13:30 – 15:50、17:55 – 18:14这几个时间段内美团app、猫眼app用户大概率不能购票

- 造成损失

预估损失出票 1,153,018 张

故障期间预估损失交易额 $1,153,018 \text{ 张} * 39.13 \text{ 元} = 45,112,238 \text{ 元}$

<https://wiki.sankuai.com/pages/viewpage.action?pageId=738045005>

17年猫眼春节故障



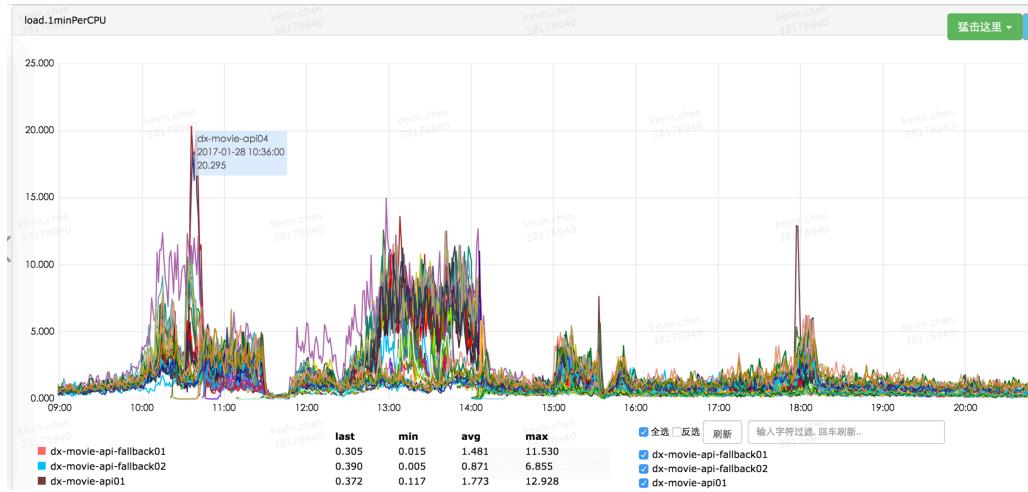
过程回顾

10:00 moviesrv开始报错，缓存出现readtimeout和connect timeout，应用负载上升

10:07 ~ 10:20 moviesrv开始扩容，两个机房共扩容10台，发布成功

10:40 页面无法正常展示，应用负载急剧上升。

10:50 ~ 10:54 打开频率流控 50次/分钟,禁10秒，10:54 通过kms打开排片页活动降级50%



17年猫眼春节故障



过程回顾

10:56 业务方发现 moviesrv的log被Medis readtimeout的异常刷屏，反馈给Medis维护同学

11:03 ~ 11:13 mmcs入口降级 MMCSFlowControl 从10000 降到0：只影响排片，不影响影院列表；只影响登录用户

11:04 - 11:17 与Medis开发及运维排查沟通客户端连接数问题，同时反馈连接超时现象。与Medis同学了解高峰期扩容的影响，11: 17敲定立刻扩容并且拒用户流量

11: 17 - 11:45 Medis替换和扩容完成 扩容了5个千兆节点

11:24 完成从nginx层面封流量 拒绝100%，返回500

11:45 ~ 12:00 Nginx流量恢复50%， mmcs入口降级 MMCSFlowControl 恢复到 5000 - 8000

12:00 mmcs负载上升，扩容mmcs集群

12:01 申请medis万兆网卡机器

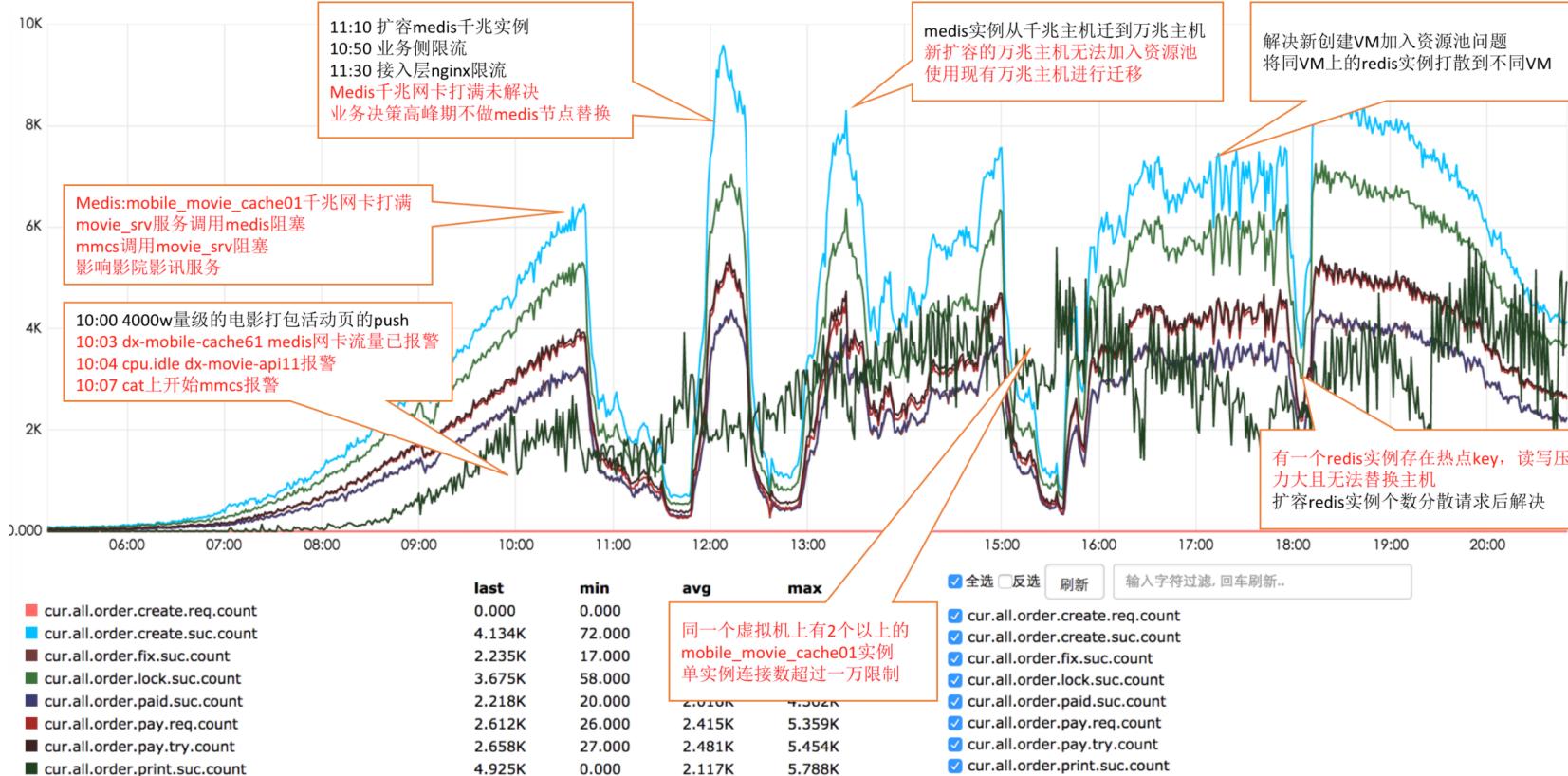
13:25~13:51 SRE NGINX 封禁全部解除，通过mmcsflow来控制，团首页猜你喜欢，查询活动标降级打开100%，各个应用集群扩容

13:51 ~ 18:00 medis机器初始化、节点替换、参数调整、继续扩容等

17年猫眼春节故障



过程回顾



17年猫眼春节故障



上海侧系统应对过程

9:17 数据库抖动，持续时间2分钟，自动恢复

9:40 DBA调整从库流量分配比例，RD调整较慢sql对应缓存的时间

10:55 DBA发现存在单条SQL QPS到8W

11:20 RD定位到改SQL走缓存的代码没生效，决定修复上线

11:35合代码准备上线，同时发现redis-movie压力大，联系王辉，发现路由策略不合理，动态调整路由策略

12:30 应用出现压力，扩容 & 重启

13:00 redis-movie依然压力大，联系王辉扩从节点

其他措施：高峰期暂停同步影讯作业、优惠查询longsql加黑名单

17年猫眼春节故障



经验教训

- 活动运营提前备案(4000w的push)
- 业务开发、SRE、基础组件三方流量预估
- 各种降级、限流能力
- 充分了解基础设施
- 业务系统去大对象(一家影院的所有影片的所有场次)

05



方法论沉淀

- 心态与意识
- 定位与决策
- 前车之鉴
- 始于足下

心态与意识



问题=机会，“坏”事变好事

积极主动，故障处理无边界

遇事不慌，与时间赛跑

你在老板心中
排第几顺位？

定位与决策



关键路径、系统指标、业务指标了然于心

通过变化找问题，自上向下，排除法

及时上升问题，寻求帮助，找到关键人物

前车之鉴



他山之石，可以攻玉

<http://issue.dp/statistics/brief>

The screenshot shows the 'Major Incident Operation Platform' interface. The left sidebar includes links for 'Incident List', 'New Incidents', 'Data Center', 'Customer End Monitoring', 'Event List', 'Timeline View', 'CASE', and 'Personal Center'. The main area displays a search bar with filters for location (Beijing), category (选择故障类别), department (选择部门), fault type (选择故障子类别), and fault level (选择故障级别). Below the search bar is a table listing incidents. The columns include: 事业群 (Business Group), 部门 (Department), 类别 (Category), 简述 (Description), 级别 (Level), 发生时间 (Occurrence Time), 定位时长 (Localization Duration), 反应时长 (Response Duration), 处理时长 (Resolution Duration), 状态 (Status), 定位方 (Localization Party), 是否通知 (Notify Status), and 操作 (Operations). One incident is highlighted in red: 技术工程及基础设施 数据 年报 20160811+,_决策报表故障 S1. Other incidents listed include: 客服平台部 客服研发 日志问题 CaseStudy-埋点log日志频繁调用引发的服务整体不可用 (影响内网S2 - 20180411) S2; 客服平台部 通讯平台_第三方 CaseStudy-20180118-10107888呼入故障 S2; 金融服务平台 支付平台 发布流程问题 CaseStudy--20171207-业务修改连接队列参数导致支付延迟 S1; 技术工程及基础设施 数据平台 基础架构 菜单中间件 CaseStudy-20171205 mlnbifit client SNAPSHOT 引发外卖订单无法查询故障 S1; 酒店旅游事业部 酒店平台 业务方的逻辑 CaseStudy-20171204-酒店拉主从延迟导致活动全量投 S1.

举一反三，问题归类

➤ 故障类型

- 线上变更：业务代码发布、中间件版本升级、SLB规则变更、线上Db库表变更、关键lion配置修改、缓存清理or配置修改、机器tomcat版本、jdk版本、jvm参数等等
- 运行时：突发流量、dns攻击、cdn节点故障等等。

➤ 常见问题

- 页面白屏，前端资源未加载
- http 请求40x、50x
- 机器负载飙升
- 频繁fullgc or outofmemory
- 网卡流量打满
- 磁盘撑爆--无空间 or 无inode数
- 线程池打满

始于足下



从我做起，从小事做起

- 熟悉业务特点
- 了解系统上下游
- 熟悉基础工具
- 专业知识查漏补缺
- 遇到问题知道找谁
- 变更及时周知大家
- 配好vpn，带上电脑



美团点评

谢

谢

聆

听