# APOLLO - 技术方案 MySQL QPS 优化

## 背景

qps 长时间在 6、7K 左右

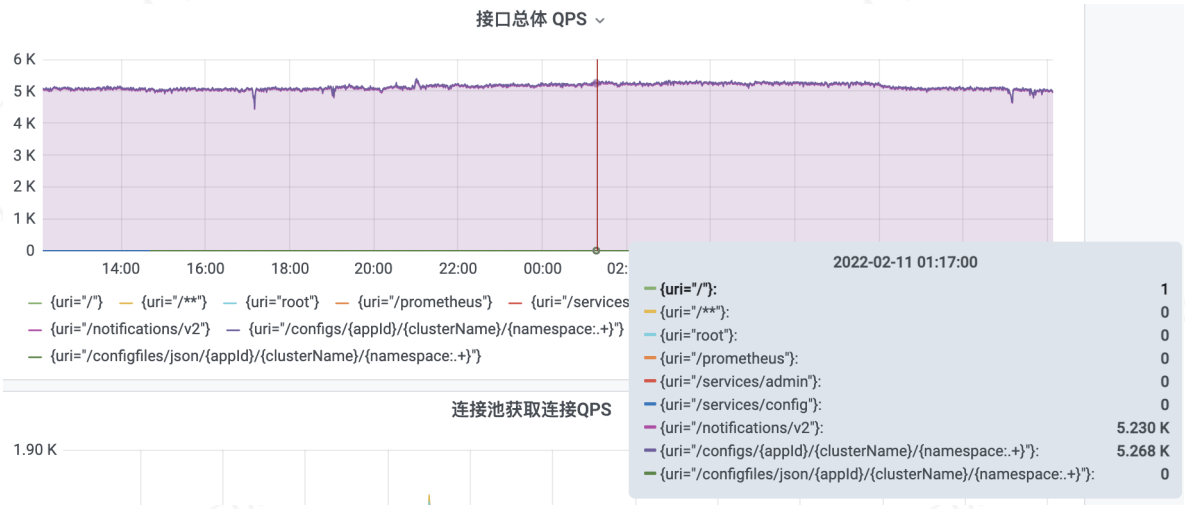
> ⚠ SPSCTP-3696 - Jira 问题不存在或者您没有权限查看。

## 解决步骤

## 服务端

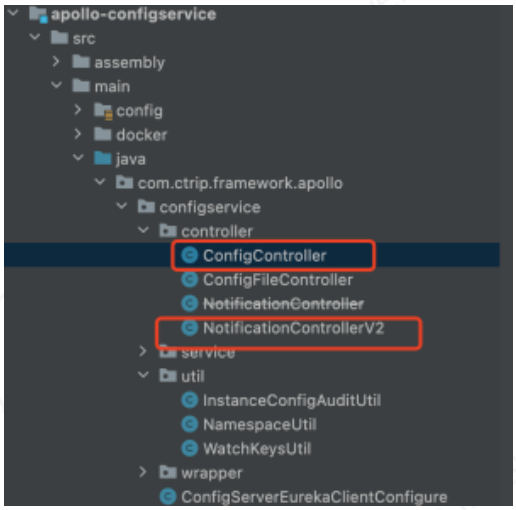
查看监控数据

https://monitoring.infra.sz.shopee.io/grafana/d/UzosPJanz/apollo?from=now-24h&orgId=7&to=now

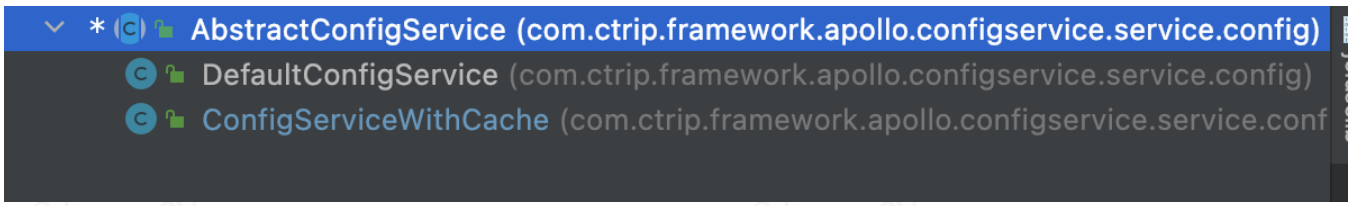



发现影响数据库qps 的主要的2个接口

查看代码位置




获取配置的核心类：




2个子类：

DefaultConfigService 直接走DB的实现

ConfigServiceWithCache 带本地缓存的实现

配置实现类：

```java
/**
 * @author Jason Song(song_s@ctrip.com)
 */
@Configuration
public class ConfigServiceAutoConfiguration {

  private final BizConfig bizConfig;

  public ConfigServiceAutoConfiguration(final BizConfig bizConfig) { this.bizConfig = bizConfig; }

  @Bean
  public GrayReleaseRulesHolder grayReleaseRulesHolder() { return new GrayReleaseRulesHolder(); }

  @Bean
  public ConfigService configService() {
    if (bizConfig.isConfigServiceCacheEnabled()) {
      return new ConfigServiceWithCache();
    }
    return new DefaultConfigService();
  }
```

```java
public boolean isConfigServiceCacheEnabled() {
  return getBooleanProperty( key: "config-service.cache.enabled",  defaultValue: false);
}
```

**live环境默认是不走cache的 ，通过数据库获取此配置**

```
# Config
# ------------------------------------------------------------
INSERT INTO `ServerConfig` (`Key`, `Cluster`, `Value`, `Comment`)
VALUES
    ('eureka.service.url', 'default', 'http://statefulset-apollo-config-server-dev-0.service-apollo-meta-server-dev:8080/eureka/,http://statefulset-apo
    ('namespace.lock.switch', 'default', 'false', '一次发布只能有一个人修改开关'),
    ('item.key.length.limit', 'default', '128', 'item key 最大长度限制'),
    ('item.value.length.limit', 'default', '20000', 'item value最大长度限制'),
    ('config-service.cache.enabled', 'default', 'false', 'ConfigService是否开启缓存，开启后能提高性能，但是会增大内存消耗！');
```

## shopee_apolloconfig_db.ServerConfig

基本信息 建表语句

筛选条件: 选择字段 操作 + Id 排序 搜索 收藏 重置

导出

| | Id | Key | Cluster | IsDeleted | DataChange_CreatedBy | DataChange_CreatedTime | DataChange_LastModifiedBy | 操作 |
|---|---|---|---|---|---|---|---|---|
| − | 5 | config-service.cache.enabled | default | | default | 2018-09-03 16:02:28 | zhiyong.liu@shopee.com | 操作历史 |

**Value:** false

**Comment:** ConfigService是否开启缓存，开启后能提高性能，但是会增大内存消耗！

如果走cache

ConfigServiceWithCache 走db的逻辑

```java
String appClusterNameLoaded = clusterName;
if (!ConfigConsts.NO_APPID_PLACEHOLDER.equalsIgnoreCase(appId)) {
  Release currentAppRelease = configService.loadConfig(appId, clientIp, appId, clusterName, namespace,
      dataCenter, clientMessages);
  if (currentAppRelease != null) {
    releases.add(currentAppRelease);
    //we have cluster search process, so the cluster name might be overridden
    appClusterNameLoaded = currentAppRelease.getClusterName();
  }
}

//if namespace does not belong to this appId, should check if there is a public configuration
if (!namespaceBelongsToAppId(appId, namespace)) {
  Release publicRelease = this.findPublicConfig(appId, clientIp, clusterName, namespace,
      dataCenter, clientMessages);
  if (!Objects.isNull(publicRelease)) {
    releases.add(publicRelease);
  }
}
```

获取配置，正常逻辑会走缓存，只有再clientMessages不为空的情况，
建造具体判断，参数message成为影响是否查询数据库的一种情况。
缓存时间为60s

获取公共配置，逻辑同上

```
//cache is out-dated
if (clientMessages != null && clientMessages.has(key) &&
    clientMessages.get(key) > cacheEntry.getNotificationId()) {
  //invalidate the cache and try to load from db again
  invalidate(key);
  cacheEntry = configCache.getUnchecked(key);
}
return cacheEntry.getRelease();
}
```

如果上报的message，key的notificationId大于缓存里面的，那么会强制走db更新缓存配置值，并返回给客户端

notifications 没有走主动查询db的逻辑

由此可知，configs接口被客户端频繁调用。

```
tcpdump port 8080 -A > /tmp/kkkk #10s
grep user-agent /tmp/kkkk |sort -n|uniq -c
```

```
root@ssconfig-configservice-58c9fc4b77-g8t8k:/tmp# grep user-agent /tmp/kkkk |sort -n|uniq -c
      4 user-agent: Go-http-client/1.1
    148 user-agent: Go-http-client/2.0
    130 user-agent: python-requests/2.14.2
     34 user-agent: python-requests/2.19.1_0
   8062 user-agent: python-requests/2.21.0
     20 user-agent: python-requests/2.26.0
root@ssconfig-configservice-58c9fc4b77-g8t8k:/tmp#
```

由图可知： 大部分python客户端频繁请求。

登录某应用服务， 发现日志频繁写入，并且频繁请求接口：

https://kubernetes.devops.i.sz.shopee.io/applications/tenants/1011/projects/fms/applications/fms-deliveryapi/deploys/fms-deliveryapi-test-my/clusters/TEST-MY:kube-general-sg2-test/pods/fms-deliveryapi-test-my-green-7b9dd75665-zxjnc?selectedTab=Terminal

grep "uncached_get" log/*

```
log/data.log:2022-02-14 16:24:19.801|uncached_get: url=https://mate-ssc.test.shopeemobile.com/configs/spx-api/my/biz.people?ip=10.103.225.142, status=200, r
sponse={"appId":"spx-api","cluster":"my","namespaceName":"biz.people","configurations":{"people.search_archive_driver_metric_flag":"True","people.async_upda
e_driver_order_operation_flag":"True","people.search_archive_driver_metric_in_tran_flag":"True","people.operation_write_flag":"2"},"releaseKey":"20210726214
26-e10539a041d17efd"}
log/data.log:2022-02-14 16:24:19.996|uncached_get: url=https://mate-ssc.test.shopeemobile.com/configs/spx-api/my/biz.zone?ip=10.103.225.142, status=200, res
onse={"appId":"spx-api","cluster":"my","namespaceName":"biz.zone","configurations":{"zone.read_heat_map_ado_config_tab_from_fms":"true","zone.cross_region_d
livery_check":"True","zone.driver_zone_station_white_list":"[\"all\"]","zone.cross_region_delivery_check_for_app":"True","zone.is_support_station_order_acco
nt":"true","zone.hub_zone_not_match_hub_level_address_list":"[\"all\"]","zone.cross_region_statistics_persistence":"True","zone.hub_zone_address_white_list"
"[\n   {\"all\":\"1\"}\n]","zone.zone_https_pool_uri_list":"[\"/api/address/match_zone\", \"/api/address/match_zone_with_attribute\", \"/api/address/zone/fe
dback\", \"/api/address/record_zone\", \"/api/address/record_driver_confidence\",\"/api/address/match_zone/v2\"]","zone.get_customized_zone_by_geo_info_flag
:"True","zone.get_text_match_threshold_by_remote_api":"False","zone.hub_zone_request_iss_time_out_ms":"3000","zone.hub_zone_white_hub_list":"[\n   {\"all\":
"1\"}\n]","zone.hub_zone_flag":"True","zone.hub_zone_match_hub_level_address_map":"{}","zone.text_match_threshold":"[]"},"releaseKey":"20220211113431-e4e22b
090817628"}
log/data.log:2022-02-14 16:24:20.264|uncached_get: url=https://mate-ssc.test.shopeemobile.com/configs/spx-api/my/biz.service_type?ip=10.103.225.142, status=
00, response={"appId":"spx-api","cluster":"my","namespaceName":"biz.service_type","configurations":{"service_type.economy_delivery_use_flag":"True","service_
type.spx_standard_use_flag":"True","service_type.groceries_delivery_use_flag":"True"},"releaseKey":"20210926113100-00baaea4dfa9363d"}
log/data.log:2022-02-14 16:24:20.550|uncached_get: url=https://mate-ssc.test.shopeemobile.com/configs/spx-api/my/biz.transport?ip=10.103.225.142, status=200
 response={"appId":"spx-api","cluster":"my","namespaceName":"biz.transport","configurations":{"transport.trigger_midway_interception_switch":"True","transpo
t.packing.fo2to.gray.country":"ID","transport.asm_to_limit":"5","transport.support.abnormal.receive":"True","transport.support_control_panel_chute_config":"
rue","transport.packing.reconciliation.batch_size":"1500","transport.support_free_status_receive":"[ALL]","transport.packing.fo2to.gray.station":"fm hub","t
ansport.trigger_handover_validation_for_3pl":"True","transport.allow_inbounding_at_manifest_level":"False","transport.asm.awb_service_switch":"True","transp
rt.asm_skip_sorting_plan_check":"True","transport.support_receive_when_pack":"[2292,2295,2296]","transport.asm_processing_record_retry_time_slice":"3600","t
ansport.awb_service_switch":"True","transport.packing.fo2to.gray.status":"0","transport.open_inter_island_flag":"false","transport.redirect_instation_flag":
False"},"releaseKey":"20220211180348-9d492b5090817668"}
root@fms-deliveryapi-test-my-green-7b9dd75665-zxjnc:/workspace#
```

分析：

正常情况， notifications 接口和 configs 接口不会太频繁。

notifications接口 没有更新时返回304. 有更新时返回200。

通过分析客户端tcpdump 日志可知，客户端收到的响应大部分是200.而不是304.

分析接收到的请求数据：

grep notifi kkkk|tail

```
.....`HGET /notifications/v2?cluster=br&notifications=%5B%7B%22notificationId%22%3A25699%2C%22namespaceName%22%3A%22lib.gray%22%7D%2C%7B%22notificationId%22
%3A%22-1%22%2C%22namespaceName%22%3A%22biz.service_type%22%7D%2C%7B%22notificationId%22%3A11839%2C%22namespaceName%22%3A%22biz.driver_wallet%22%7D%2C%7B%22no
tificationId%22%3A22471%2C%22namespaceName%22%3A%22biz.delivery%22%7D%2C%7B%22notificationId%22%3A12588%2C%22namespaceName%22%3A%22biz.dashboard%22%7D%2C%7B%
22notificationId%22%3A14941%2C%22namespaceName%22%3A%22biz.people%22%7D%2C%7B%22notificationId%22%3A27604%2C%22namespaceName%22%3A%22lib%22%7D%2C%7B%22notifi
cationId%22%3A12435%2C%22namespaceName%22%3A%22biz.zone%22%7D%2C%7B%22notificationId%22%3A24430%2C%22namespaceName%22%3A%22application%22%7D%2C%7B%22notifica
tionId%22%3A25698%2C%22namespaceName%22%3A%22biz.station%22%7D%2C%7B%22notificationId%22%3A22055%2C%22namespaceName%22%3A%22biz.order%22%7D%2C%7B%22notificat
ionId%22%3A25587%2C%22namespaceName%22%3A%22biz.order_tracking%22%7D%2C%7B%22notificationId%22%3A28179%2C%22namespaceName%22%3A%22biz.address%22%7D%2C%7B%22n
otificationId%22%3A26483%2C%22namespaceName%22%3A%22biz.pickup%22%7D%2C%7B%22notificationId%22%3A24434%2C%22namespaceName%22%3A%22biz.order_recipient%22%7D%2
C%7B%22notificationId%22%3A25229%2C%22namespaceName%22%3A%22biz.account%22%7D%2C%7B%22notificationId%22%3A27185%2C%22namespaceName%22%3A%22biz.transport%22%7
D%2C%7B%22notificationId%22%3A24433%2C%22namespaceName%22%3A%22biz.finance%22%7D%2C%7B%22notificationId%22%3A8223%2C%22namespaceName%22%3A%22biz.p2p%22%7D%2C
%7B%22notificationId%22%3A25165%2C%22namespaceName%22%3A%22biz.workforce%22%7D%2C%7B%22notificationId%22%3A25765%2C%22namespaceName%22%3A%22biz.3pl%22%7D%2C%
7B%22notificationId%22%3A26449%2C%22namespaceName%22%3A%22biz.delivery_order%22%7D%2C%7B%22notificationId%22%3A24435%2C%22namespaceName%22%3A%22biz.returning
%22%7D%2C%7B%22notificationId%22%3A26855%2C%22namespaceName%22%3A%22biz.fleet_order%22%7D%2C%7B%22notificationId%22%3A24436%2C%22namespaceName%22%3A%22biz.ti
cket%22%7D%2C%7B%22notificationId%22%3A20203%2C%22namespaceName%22%3A%22biz.dropoff%22%7D%2C%7B%22notificationId%22%3A14465%2C%22namespaceName%22%3A%22biz.dr
iver%22%7D%5D&appId=spx-api HTTP/1.1
```

[{"notificationId":25699,"namespaceName":"lib.gray"},{"notificationId":"-1","namespaceName":"biz.service_type"},

{"notificationId":11839,"namespaceName":"biz.driver_wallet"},{"notificationId":22471,"namespaceName":"biz.delivery"},

{"notificationId":12588,"namespaceName":"biz.dashboard"},{"notificationId":14941,"namespaceName":"biz.people"},{"notificationId":27604,"namespaceName":"lib"},

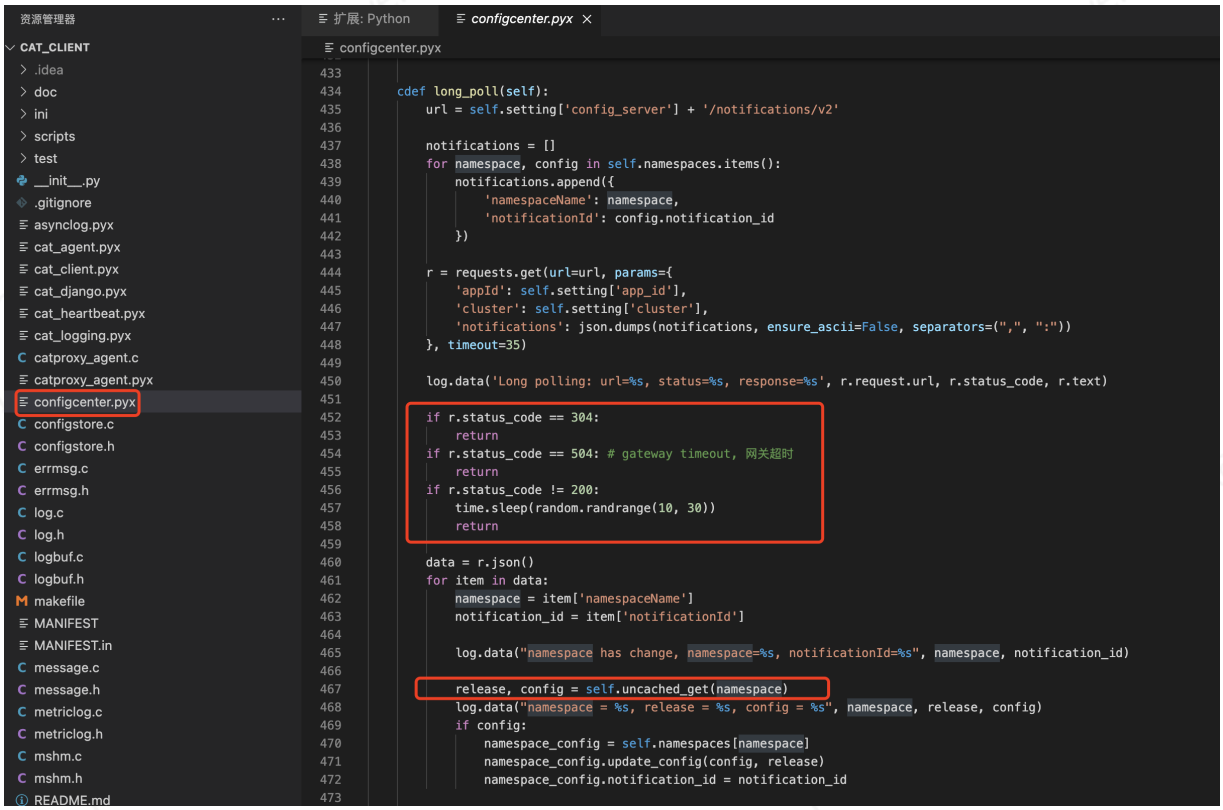
{"notificationId":12435,"namespaceName":"biz.zone"},{"notificationId":24430,"namespaceName":"application"},{"notificationId":25698,"namespaceName":"biz.station"},

{"notificationId":22055,"namespaceName":"biz.order"},{"notificationId":25587,"namespaceName":"biz.order_tracking"},

{"notificationId":28179,"namespaceName":"biz.address"},{"notificationId":26483,"namespaceName":"biz.pickup"},

{"notificationId":24434,"namespaceName":"biz.order_recipient"},{"notificationId":25229,"namespaceName":"biz.account"},

阅读python 代码发现 当 返回为200时，会一直调用configs接口(ucached_get)




分析接口返回的数据可知：

当configurations:"{}"

也就是发布的配置为空时：

```
def uncached_get(self, namespace='application'):
    url = '%s/configs/%s/%s/%s' % \
          (self.setting['config_server'], self.setting['app_id'], self.setting['cluster'], namespace)

    if self.ip:
        url = url + '?ip=' + self.ip

    r = requests.get(url)

    log.data('uncached_get: url=%s, status=%s, response=%s', r.request.url, r.status_code, r.text)

    if r.ok:
        data = r.json()
        return data['releaseKey'], data['configurations']

    return None, None
```

”if config“（判断排除了包含空对象和 None）的逻辑永远不会进去，因此 notificationId 不会被更新，造成不断获取新数据

```
data = r.json()
for item in data:
    namespace = item['namespaceName']
    notification_id = item['notificationId']

    log.data("namespace has change, namespace=%s, notificationId=%s", namespace, notification_id)

    release, config = self.uncached_get(namespace)
    log.data("namespace = %s, release = %s, config = %s", namespace, release, config)
    if config:
        namespace_config = self.namespaces[namespace]
        namespace_config.update_config(config, release)
        namespace_config.notification_id = notification_id
```

赋值

解决方案： 修改 client 的判断逻辑 改为 "not config is None"