# 监控告警指标定义

# 指标定义

## 访问量

### 突发流量异常请求

#### 说明

判断检测时间点指定时间范围内，最近四周同时间的访问量(每秒)环比平均值，设置一个最低告警范围和异常范围，计算出告警范围的最高值，如果当前时间范围内的平均值高于最低告警值，则触发超量异常请求告警。

#### 缺陷

未考虑特殊时间影响，如：节假日影响因素；以及各类促销活动的影响因素，如：抽奖、派红包等引起的流量突增。

#### 计算方式



#### 变量定义

采样时间范围(Ts)： 15分钟

告警范围（W）： 根据访问量大小动态计算

异常范围（E）： 根据访问量大小动态计算

**告警范围计算方法：**

访问量小于5：默认为43%

访问量大于5：

**异常范围计算方法：**

访问量小于5：默认为83%

访问量大于5：

#### 适用范围

适用于所有应用程序及接口流量的告警监控。前期优先实现对程序流量的告警监控，对于接口的告警监控，可根据接口的重要程度适量增加。

### 过低异常请求

#### 说明

判断检测时间点指定时间范围内，最近四周同时间的访问量(每秒)环比平均值，设置一个最低告警范围和异常范围，计算出告警范围的最低值，如果当前时间范围内的平均值低于最低告警值，则触发过低异常请求告警。

#### 缺陷

暂无。

#### 计算方式



#### 变量定义

采样时间范围(Ts)： 15分钟

告警范围（W）： 根据访问量大小动态计算

异常范围（E）： 根据访问量大小动态计算

**告警范围计算方法：**

访问量小于5：默认为43%

访问量大于5：

**异常范围计算方法：**

访问量小于5：默认为83%

访问量大于5：

#### 适用范围

适用于所有应用程序及接口流量的告警监控。前期优先实现对程序流量的告警监控，对于接口的告警监控，可根据接口的重要程度适量增加。

## 响应时间

#### 说明

对于响应时间来说，一般情况平均值会维持在某一个值上下浮动，不会出现太大的波动，因此不需要针对不同时间点的范围来取样进行调整，只需要监控指定时间段的响应时间平均值即可，每个应用或接口，可以取某一天的平均响应时间作为参考值，当采样时间范围的平均响应时间大于参考值的最大警告范围时，则触发报警。

#### 缺陷

暂无。

#### 计算方式



#### 变量定义

采样时间范围(Ts)： 15分钟（如请求数量不足100，则加大采样时间范围）

告警范围（W）： 根据访问量大小动态计算

异常范围（E）： 根据访问量大小动态计算

**告警范围计算方法：**

响应时间小于5：默认为43%

响应时间大于5：

**异常范围计算方法：**

响应时间小于5：默认为83%

响应时间大于5：

#### 适用范围

适用于所有应用程序及接口流量的告警监控。前期优先实现对程序流量的告警监控，对于接口的告警监控，可根据接口的重要程度适量增加。

## 异常码

### 业务异常

#### 说明

业务异常监控HTTP状态码返回状态为4XX的请求数。一般来说，基于RESTful规范的应用，会有一定比例的4XX请求，因此每个应用对于业务的异常码来说，其告警或异常的比例会有一定的差异，可由监控对象自行定义。

在本次指标中，4XX类请求仅提供告警通知。

#### 缺陷

暂无。

#### 计算方式



#### 变量定义

采样时间范围(T)： 15分钟

告警占比（Rw）： 30%

异常占比（Re）： 50%

样本值数量(N)： 100

#### 适用范围

适用于所有应用程序及接口流量的告警监控。前期优先实现对程序流量的告警监控，对于接口的告警监控，可根据接口的重要程度适量增加。

### 服务异常

#### 说明

服务异常监控HTTP状态码返回状态为5XX的请求数。如果程序出现了较多的5XX状态码，一般情况下，返回5XX表示程序内部出现了一些异常，因此这个值的监控范围需要缩小，建议：告警占比为15%，异常占比为：30%。

#### 缺陷

暂无。

#### 计算方式



#### 变量定义

采样时间范围(T)： 15分钟

告警占比（Rw）： 15%

异常占比（Re）： 30%

样本值数量(N)： 100

#### 适用范围

适用于所有应用程序及接口流量的告警监控。前期优先实现对程序流量的告警监控，对于接口的告警监控，可根据接口的重要程度适量增加。

# 监控架构

## 实现方式

基于现有的Nagios或Zabbix系统，使用Python脚本编写检测脚本，从ES中查询监控数据并分析，返回是否需要通知的结果到监控系统，由监控系统再进行告警通知。

Python脚本需要注意如下事项：

1. 脚本运行由Nagios/Zabbix系统配置并调度
2. 脚本需要考虑重复通知的问题，即：出现异常的时候，一般相当长时间内都会有异常，因此，需要记录初次通知的状态以及恢复时的状态，在异常期内，除非数据发生严重变更，则只通知两次（初次异常和异常恢复）
3. 脚本需要考虑频繁调用的问题，假设脚本执行过慢，在下次调度的时上次脚本还未执行完成，应该主动忽略本次调用

## 通知流程

