事件队列

CPU

网络

1-req

2-req

3-req

1-exec

druration 1

1-rep

2-exec

druration 2

2-exec

druration 3

2-rep

3-rep

1-send

druration 1s

2-send

druration 2s

3-send

druration 3s

以上图为例进行说明

上图假设同一时间，服务器接到1、2、3这3个请求

事件队列部分从上到下，依次是事件进入队列的顺序。

druration 1、druration 2、druration 3分别是请求1、2、3的执行时间，当执行时间超过Redis设置的slow log阈值，将会被写入文件。不过这3个时间不是请求在服务器的驻留时间。

驻留时间这里指从请求达到服务器到被处理后返回到客户端的时间

请求1在服务器的驻留时间是druration 1+ druration 2+ druration 3，因为请求1的消息返回事件在队列中被放置到事件3的处理之后，只有当事件3被处理结束后，才进行消息的返回。

请求2在服务器的驻留时间为druration 1+ druration 2+ druration 3+ druration 1s，这是因为事件2的消息返回要等待事件1的消息发送完毕

请求的情况以此类推

因此，会出现这样一种情况，客户端记录了执行时间较长的redis请求，但服务器却没有任何slow log的日志被记录。

为了更好的了解请求的处理情况，请求驻留在服务器时，我们将其再分为两个时间点

时间点1：从接受请求到请求被处理

时间点2：从接受请求到请求被发送到客户端

前者可以了解等待的情况，后者可以了解在整体运行中，在服务器的驻留情况

在ctrip的redis中增加了对这二者和执行时间的均值统计，如下图，分别是命令执行时间的均值统计、时间点1和时间点2的均值统计

