

P95、P99.9百分位数值——服务响应时间的重要衡量指标

原创 老於记 于 2021-02-01 08:11:00 发布 5087 收藏 14

文章标签: [java](#) [python](#) [数据分析](#) [大数据](#) [人工智能](#)

前段时间,在对系统进行改版后,经常会有用户投诉说页面响应较慢,我们看了看监控数据,发现从接口响应时间的平均值来看在500ms左右,也算符合要求,不至于像用户说的那么慢,岁很费解,后来观察其它的一些指标发现确实是有问题,这个指标就是P95、P99.9,我们发现虽然平均响应时间并不高,但P95和P99.9却达到了2s以上,说明我们的接口确实存在慢查询。于是捞取了一些慢查询的请求日志终于发现问题。那么P95、P99又代表什么意思呢?

通常,我们对服务响应时间的衡量指标有Min(最小响应时间)、Max(最大响应时间)、Avg(平均响应时间)等。

1 平均值Avg

其中比较常用的值就是平均值,例如平均耗时为100ms,表示服务器当前请求的总耗时/请求总数量,通过该值,我们大体能知道服务运行情况。

但是使用平均值来衡量响应时间有个非常大的问题,举个例子:众所周知,我和Jack马和tony马的财富加起来足以撼动整个亚洲,我和姚明的平均身高有两米多.....

平均值同样有这种问题,这个衡量指标的计算方式会把一些异常的值平均掉,进而会掩盖一些问题,我们只知道所有请求的平均响应时间是100ms,但是具体有多少个请求比100ms要大,又有多少个请求比100ms要小,大多少,是200ms,还是500ms,又或是1000ms,我们无从得知。

2 百分位数值

平均值并不能反映数据分布及极端异常值的问题,这时我们可以使用百分位数值。

百分位数值是一个统计学中的术语。

如果将一组数据从小到大排序,并计算相应的累计百分位,则某一百分位所对应数据的值就称为这一百分位的百分位数。可表示为:一组n个观测值按数值大小排列。如,处于p%位置的值称第p百分位数

用我们软件开发行业的例子通俗来讲就是,假设有100个请求,按照响应时间从小到大排列,位置为X的值,即为PX值。

P1就是响应时间最小的请求,P10就是排名第十的请求,P100就是响应时间最长的请求。

在真正使用过程中,最常用的主要有P50(中位数)、P95、P99。

P50: 即中位数值。100个请求按照响应时间从小到大排列,位置为50的值,即为P50值。如果响应时间的P50值为200ms,代表我们有半数的用户响应耗时在200ms之内,有半数的用户响应耗时大于200ms。如果你觉得中位数值不够精确,那么可以使用P95和P99.9

P95: 响应耗时从小到大排列,顺序处于95%位置的值即为P95值。

还是采用上面那个例子,100个请求按照响应时间从小到大排列,位置为95的值,即为P95值。我们假设该值为200ms,那这个值又表示什么意思呢?

意思是说,我们对95%的用户的响应耗时在200ms之内,只有5%的用户的响应耗时大于200ms,据此,我们掌握了更精确的服务响应耗时信息。

P99.9: 许多大型的互联网公司会采用P99.9值,也就是99.9%用户耗时作为指标,意思就是1000个用户里面,999个用户的耗时上限,通过测量与优化该值,就可保证绝大多数用户的使用体验。至于P99.99值,优化成本过高,而且服务响应由于网络波动、系统抖动等不能解决之情况,因此大多数时候都不考虑该指标。

下图是我从我们系统中随便拉的两个接口的性能监控数据,我们可以看到第一个均值在40ms,P95在82.5ms,看似还可以,但是P99.9却是1743ms。

而第二个接口均值在710ms,但是P95却是1592.7ms,这代表我们有将近5%的用户访问该接口的时间要大于1592.7ms。P99.9更是达到了2494.2ms。

Count	Sum	Self	Failure	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	SAvg(ms)	P95(ms)	P99.9(ms)	QPS	Percent%
64,653	43.1m	42.5m	0	0.0000%	[:: show ::]	0	3,855	40.0	39.5	82.5	1,743.0	19.7	100.00%
1,973	23.3m	23.3m	0	0.0000%	[:: show ::]	63	3,855	710.0	708.3	1,592.7	2,494.2	0.6	3.05%

以上两个接口如果单纯只看均值指标，并没有什么问题，但是P95和P99.9却反映了我们一些慢请求的情况。拿到这个指标数据，我们就知道我们的服务并非没有问题，就可以去优化这两个指标的值，以达到更好的用户体验。

3 如何计算百分位数

平均值之所以会成为大多数人使用衡量指标，其原因主要在于他的计算非常简单。请求的总耗时/请求总数量就可以得到平均值。而P值的计算则相对麻烦一些。

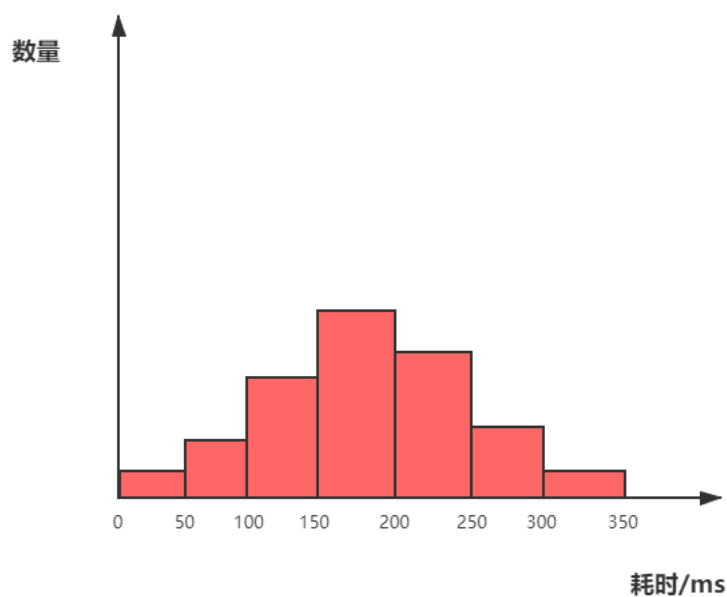
按照传统的方式，计算P值需要将响应耗时从小到大排序，然后取得对应百分位之值。

如果服务qps较低，例如：100/秒，我们计算这1s内的P值，就记录这100请求的耗时数据，然后排序，然后取得P分位值，并非难事。但如果我们要计算1h内的p值呢，就是要对360000的数据进行排序然后取得P分位值。而如果对于一些用户量更大的系统，例如：QPS 30万/秒，那么1h内的p值如果还是采用记录+排序的方式，就是要对十个多亿的数据进行排序，可想而知需要消耗多么大的内存与计算资源。

那么有没有简单的计算方式呢？

可以采用分桶计算的方式，即一个耗时范围一个桶，该计算方式虽不是完全准确值，但精度非常高，误差较小。

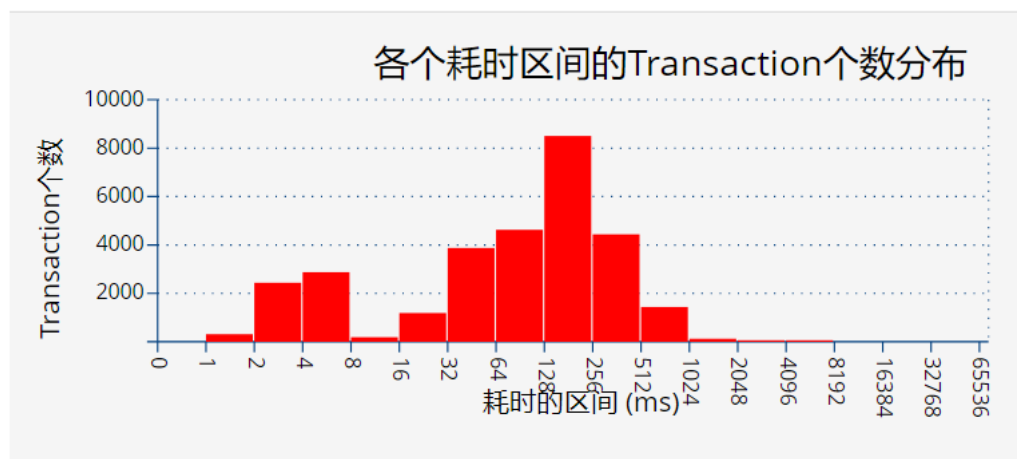
首先需要界定每个桶的跨度，可以采用等分形式，例如对于耗时统计需求，我们可以假定一个耗时上界，然后等分成N个区间，如下图，如果响应耗时在30ms则落在0-50ms的桶内，如果响应时间在80ms则落在50-100ms的桶内，以此类推。



这样就避免了对全部数据进行排序，只需要根据各个桶中的数据数量，即可计算出95%位置位于哪个桶，例如需要计算95线时，就从最大的桶开始剔除，当数量超过5%的时候，那个桶的值就是95线。然后在桶的内部采用插值方法，也可以通过桶内平均的方式来计算出一个相对精确的P95值。

此外，考虑到数据分布特点，服务耗时异常数据应该只是少数，但是异常值跨度可能很大，大部分耗时数据均靠近正常值，如果采用桶等分的形式，可能会导致大量数据堆积在一个桶内中，又如何解决这个问题？

其实可以采用非等分的跨度划分方式，例如采用**指数**形式划分，耗时越低的区间，跨度越小，精度约高。



此外也可以采用美团点评的实时监控系统cat的桶跨度划分方式，代码如下：

```
1 public static int computeDuration(int duration) {
2     if (duration < 1) {
3         return 1;
4     } else if (duration < 20) {
5         return duration;
6     } else if (duration < 200) {
7         return duration - duration % 5;
8     } else if (duration < 500) {
9         return duration - duration % 20;
10    } else if (duration < 2000) {
11        return duration - duration % 50;
12    } else if (duration < 20000) {
13        return duration - duration % 500;
14    } else if (duration < 100000) {
15        return duration - duration % 10000;
16    } else {
17        int dk = 524288;
18
19        if (duration > 3600 * 1000) {
20            dk = 3600 * 1000;
21        } else {
22            while (dk < duration) {
23                dk <<= 1;
24            }
25        }
26        return dk;
27    }
28 }
```

即：小于20ms的时候1ms一个桶，大于20ms小于200ms的时候5ms一个桶，大于200ms小于500ms的时候20ms一个桶，以此类推！而桶的值也可以作为百分位数的近似值，而无需进行排序计算，这个时候约耗时越小的时候，精度越准确！

小结

百分位数值在互联网系统中有很大的意义。**通过对百分位数值的监控与优化，我们可以将更多的用户纳入我们的监控体系中，让我们的服务能够对绝对大多数的用户提供更好的体验！** 在一些错误率、异常率上面我们也可以使用百分位数来进行系统可用性是否达到要求，甚至在一些新的产品特性或者AB测试上也可以用来统计分析用户对其的反响，以此来衡量该特性是否真正对用户有帮助.....