# 第一性能测试几个重要计算公式

## 1、响应时间：对请求作出响应所需要的时间

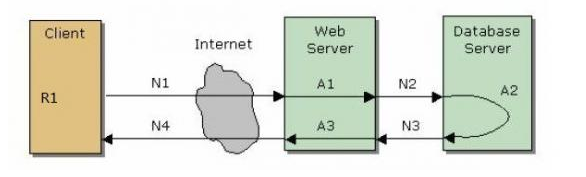
网络传输时间：N1+N2+N3+N4

应用服务器处理时间：A1+A3

数据库服务器处理时间：A2

响应时间=N1+N2+N3+N4+A1+A3+A2

其中：N1+N2+N3+N4+A1+A3+A2如下图所示



## 2、并发用户数的计算公式

系统用户数：系统额定的用户数量，如一个OA系统，可能使用该系统的用户总数是5000个，那么这个数量，就是系统用户数。

同时在线用户数：在一定的时间范围内，最大的同时在线用户数量。  
同时在线用户数=每秒请求数RPS（吞吐量）+并发连接数+平均用户思考时间

平均并发用户数的计算：C=nL / T

其中**C是平均的并发用户数**，**n是平均每天访问用户数（login session）**，**L是一天内用户从登录到退出的平均时间（login session的平均时间）**，**T是考察时间长度（一天内多长时间有用户使用系统）**

并发用户数峰值计算：C^约等于C + 3\*根号C

其中C^是并发用户峰值，C是平均并发用户数，该公式遵循泊松分布理论。

**C^也是我们系统要求的最大并发数量；**

## 3、吞吐量的计算公式

指单位时间内系统处理用户的请求数

从业务角度看，吞吐量可以用：请求数/秒、页面数/秒、人数/天或处理业务数/小时等单位来衡量

从网络角度看，吞吐量可以用：**字节/秒来衡量**

对于交互式应用来说，吞吐量指标反映的是服务器承受的压力，他能够说明系统的负载能力

以不同方式表达的吞吐量可以说明不同层次的问题，例如，以字节数/秒方式可以表示数要受网络基础设施、服务器架构、应用服务器制约等方面的瓶颈；已请求数/秒的方式表示主要是受应用服务器和应用代码的制约体现出的瓶颈。

**当没有遇到性能瓶颈的时候**，吞吐量与虚拟用户数之间存在一定的联系，可以采用以下公式计算：F=VU \* R /T

其中

F为吞吐量；

VU：LR脚本中设置的用户数；

R：每个用户请求的个数；

T：测试时长；

**因此要根据测试报告中的吞吐量F与VU 、R、T直接是否存在F=VU \* R /T关系来判断是否存在性能瓶颈；**

## 4、性能计数器

是描述服务器或操作系统性能的一些数据指标，如使用内存数、进程时间，在性能测试中发挥着“监控和分析”的作用，尤其是在分析统统可扩展性、进行新能瓶颈定位时有着非常关键的作用。

资源利用率：指系统各种资源的使用情况，如cpu占用率为68%，内存占用率为55%，一般使用“资源实际使用/总的资源可用量”形成资源利用率。

## 5、思考时间的计算公式

Think Time，从业务角度来看，这个时间指用户进行操作时每个请求之间的时间间隔，而在做新能测试时，为了模拟这样的时间间隔，引入了思考时间这个概念，来更加真实的模拟用户的操作。

在吞吐量这个公式中F=VU \* R / T说明吞吐量F是VU数量、每个用户发出的请求数R和时间T的函数，而其中的R又可以用时间T和用户思考时间TS来计算：R = T / TS

下面给出一个计算思考时间的一般步骤：

A、首先计算出系统的并发用户数

C=nL / T F=R×C

B、统计出系统平均的吞吐量

F=VU \* R / T R×C = VU \* R / T

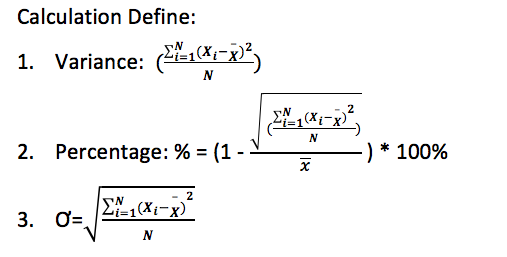
C、统计出平均每个用户发出的请求数量

R=u\*C\*T/VU

D、根据公式计算出思考时间

TS=T/R

## 6系统稳定性的计算公式

****

方差：

平均数

变异系数：

变异系数越小，变异(偏离)程度越小，风险也就越小；反之，变异系数越大，变异(偏离)程度越大，风险也就越大。

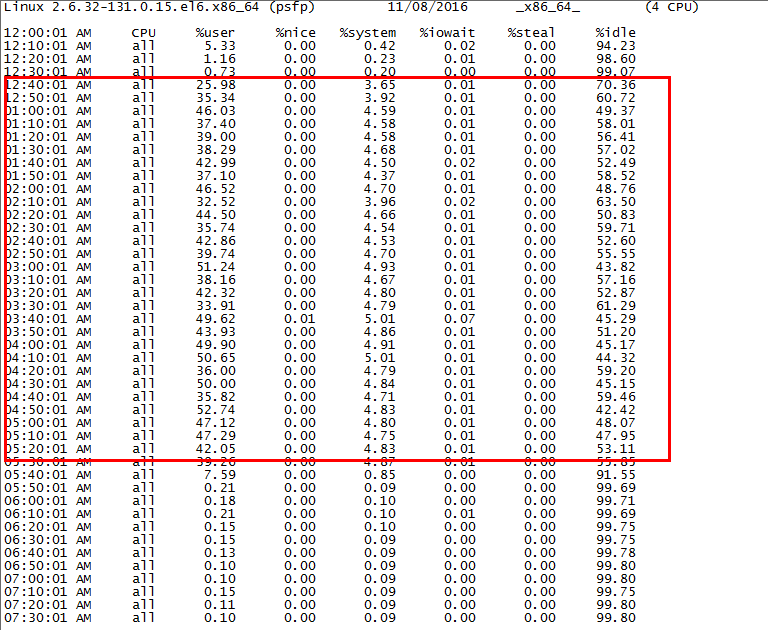
那么系统中计算稳定性：100%-C.V

关于系统需要计算哪些稳定性：

1. 吞吐量；2、CPU利用率；3、TPS； 4、服务器磁盘IO ；5、响应时间

如何取值：**top取的是瞬时值，sar取的是平均值；**所以采用sar的值作为稳定性的取值点；

如CPU利用率的稳定性计算；脚本运行时间为：12:40：01 AM到05：30:01AM



那么计算CPU的稳定性为：

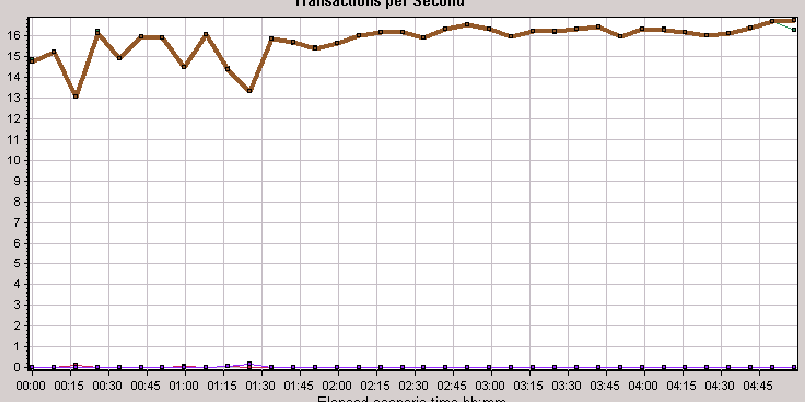
先求平均:=[（25.98+3.65）+(35.34+3.92)+(46.03+4.59)+(37.40+4.58)+(39.00+4.58)+(38.29+4.50)+(42.99+4.50)+(37.10+4.37)+(46.52+4.70)+(32.52+3.96)+(44.50+4.66)+(35.74+4.54)+(42.86+4.53)+（39.74+4.70）+（51.24+4.93）+(38.16+4.67)+(42.32+4.80)+(33.91+4.79)+(49.62+5.01)+（43.93+4.86）+（49.90+4.91）+（50.65+5.01）+（36.00+4.79）+（50.00+4.84）+（35.82+4.71）+（52.74+4.83）+（47.12+4.80）+（47.29+4.75）+（42.05+4.83）+（39.26+4.87）]/30=46.44

然后求方差：=44

标准差：S==6.6332495807107996982298654733414

稳定性：（1-）100=85.72%

那TPS,这么取值如下图所示：



采用取取拐点方式

# 第二、软件性能的关注点

## 1、用户角度：

响应时间

## 2、管理员的角度：

1、响应时间  
2、服务器资源使用情况是否合理  
3、 应用服务器和数据库资源使用是否合理  
4、 系统能否实现扩展  
5、 系统最多支持多少用户访问、系统最大业务处理量是多少  
6、 系统性能可能存在的瓶颈在哪里  
7、 更换那些设备可以提高性能  
8、 系统能否支持7×24小时的业务访问

## 3、开发人员角度

1、 架构设计是否合理  
2、 数据库设计是否合理  
3、 代码是否存在性能方面的问题  
4、 系统中是否有不合理的内存使用方式  
5、 系统中是否存在不合理的线程同步方式  
6、 系统中是否存在不合理的资源竞争

# 第三、性能测试主要概念

## **一．系统吞度量要素：**

一个系统的吞度量（承压能力）与request对CPU的消耗、外部接口、IO等等紧密关联。单个reqeust 对CPU消耗越高，外部系统接口、IO影响速度越慢，系统吞吐能力越低，反之越高。系统吞吐量几个重要参数：QPS（TPS）、并发数、响应时间

**QPS（TPS）：**每秒钟request/事务 数量

**并发数：** 系统同时处理的request/事务数

**响应时间：** 一般取平均响应时间

解了上面三个要素的意义之后，就能推算出它们之间的关系：

QPS（TPS）= 并发数/平均响应时间

        一个系统吞吐量通常由QPS（TPS）、并发数两个因素决定，每套系统这两个值都有一个相对极限值，在应用场景访问压力下，只要某一项达到系统最高值，系统的吞吐量就上不去了，如果压力继续增大，系统的吞吐量反而会下降，原因是系统超负荷工作，上下文切换、内存等等其它消耗导致系统性能下降。决定系统响应时间要素；我们做项目要排计划，可以多人同时并发做多项任务，也可以一个人或者多个人串行工作，始终会有一条关键路径，这条路径就是项目的工期。

系统一次调用的响应时间跟项目计划一样，也有一条关键路径，这个关键路径是就是系统影响时间；关键路径是有CPU运算、IO、外部系统响应等等组成。

## **二．系统吞吐量评估：**

我们在做系统设计的时候就需要考虑CPU运算、IO、外部系统响应因素造成的影响以及对系统性能的初步预估。

而通常境况下，我们面对需求，我们评估出来的出来QPS、并发数之外，还有另外一个维度：日PV。

通过观察系统的访问日志发现，在用户量很大的情况下，各个时间周期内的同一时间段的访问流量几乎一样。比如工作日的每天早上。只要能拿到日流量图和QPS我们就可以推算日流量。

通常的技术方法：

        1. 找出系统的最高TPS和日PV，这两个要素有相对比较稳定的关系（除了放假、季节性因素影响之外）

        2. 通过压力测试或者经验预估，得出最高TPS，然后跟进1的关系，计算出系统最高的日吞吐量。

# 第四：我们报告需要增加的点