



# 软件性能测试实践

陈 雷

[Jackeichan@gmail.com](mailto:Jackeichan@gmail.com)

<http://Jackei.cnblogs.com>

# 内容提要

- ◆理解“性能”与“性能测试”
- ◆性能测试过程 与 最佳实践
- ◆相关主题讨论

# 内容提要

## ◆理解“性能”与“性能测试”

- 什么是性能测试
- 如何评价一个系统的性能
- 理发店模型

## ◆性能测试的过程 与 实践

## ◆相关主题讨论

# 什么是性能测试

- ◆观察系统在一个给定的环境和场景中的性能表现是否与预期目标一致，评判系统是否存在性能缺陷，并根据测试结果识别性能瓶颈，改善系统性能的完整的过程

# 功能测试 vs. 性能测试

	功能测试	性能测试
测试目的	1.验证系统是否满足需求 2.查找系统中可能存在的功能缺陷	1.验证系统是否满足需求 2.查找系统中可能存在的性能缺陷或瓶颈
测试依据	系统的功能需求	系统的性能需求
测试过程	分析系统需求—>设计测试用例模拟用户场景—>执行测试—>提交缺陷—>验证缺陷是否解决	分析系统需求—>设计性能测试用例模拟用户场景—>执行测试—>提交缺陷—>验证缺陷是否解决
行业背景	要求对业务和用户场景非常熟悉	要求对业务和用户场景非常熟悉

# 内容提要

## ◆理解“性能”与“性能测试”

- 什么是性能测试
- 如何评价一个系统的性能
- 理发店模型

## ◆性能测试的过程 与 实践

## ◆相关主题讨论

# 如何评价系统的性能

## ◆ 用户(end-user)的视角

- 响应时间(Response Time)

## ◆ 运营商(customer)和开发商的视角

- 响应时间(Response Time)
- 并发用户数(The Number of Concurrent Users)
- 吞吐量(Throughput) – 每秒交易数(Transaction per Second)
- 资源利用率(Hardware/Software Resource Utilization)
- 可靠性或稳定性(Reliability or Stability)
- 可伸缩性(Scalability)
- 可恢复性(Recoverability)

# 内容提要

## ◆理解“性能”与“性能测试”

- 什么是性能测试
- 如何评价一个系统的性能
- 理发店模型

## ◆性能测试的过程 与 实践

## ◆相关主题讨论





WWW.GZTEST.NET

# 理发店模型的 3 个假设

- ◆ 理发店中一共有 3 名理发师
- ◆ 每位理发师剪一个发的时间都是 1 小时
- ◆ 我们顾客们都是很有时间观念的人而且非常挑剔，他们对于每次光顾理发店时所能容忍的等待时间+剪发时间是3小时，而且等待时间越长，顾客的满意度越低。如果3个小时还不能剪完头发，我们的顾客会立马生气的走人

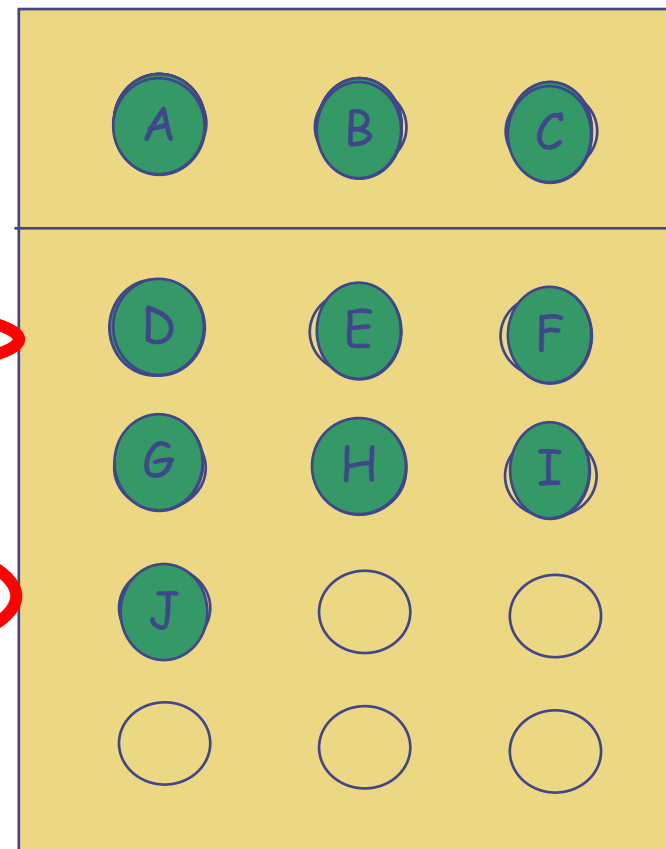




WWW.GZTEST.NET

# 理发店内场景的模拟

顾客数	理发师状态	顾客状态
1	有两名理发师空闲	顾客不需要等待
2	有一名理发师空闲	顾客不需要等待
3	没有理发师空闲	没有顾客需要等待
4	没有理发师空闲	有1名顾客需要等待
9	没有理发师空闲	接近顾客所能忍受的最大限度，顾客满意度接近最低
10	没有理发师空闲	必然有一位顾客最终会选择离开

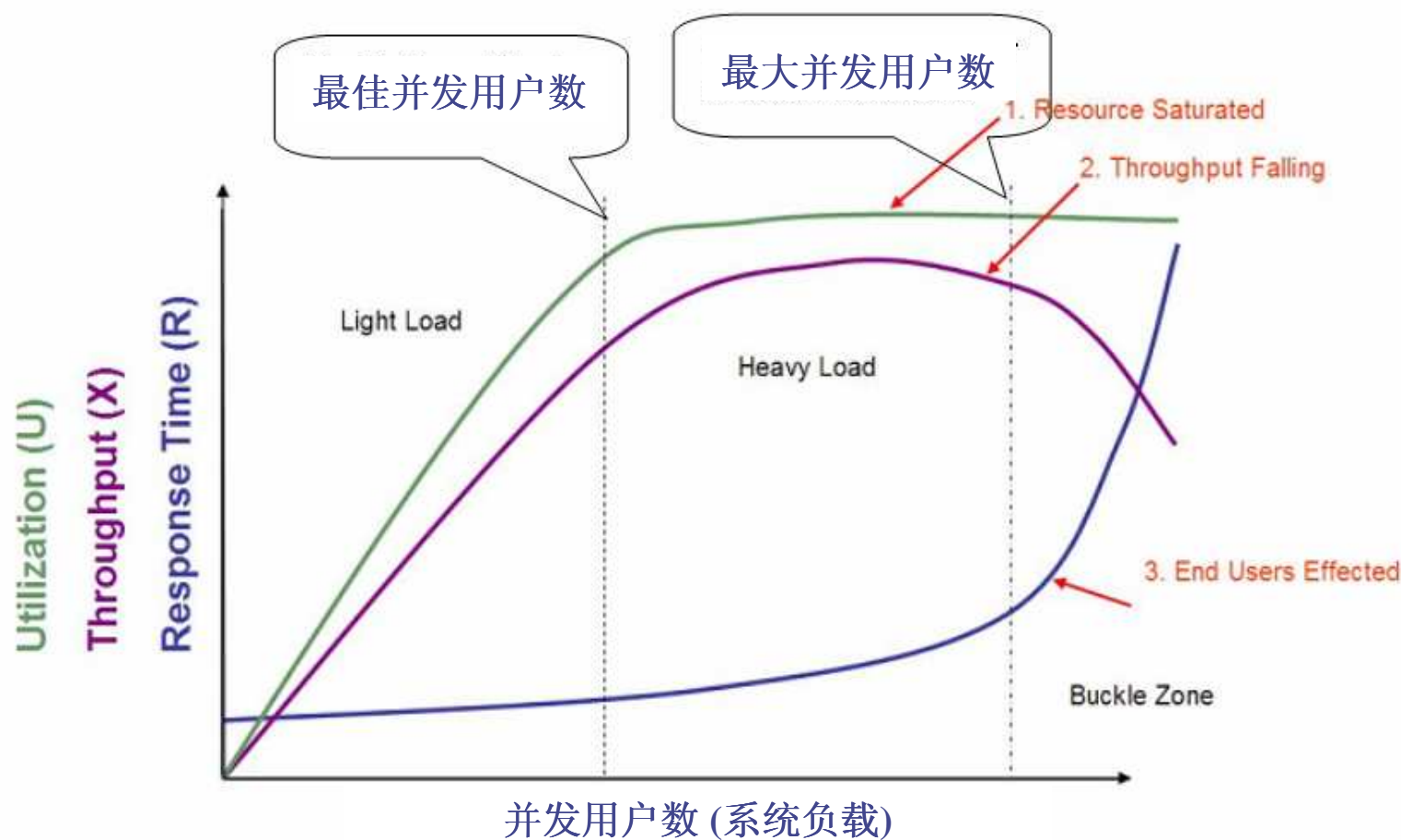


如果持续 每个小时>3个 顾客？



WWW.GZTEST.NET

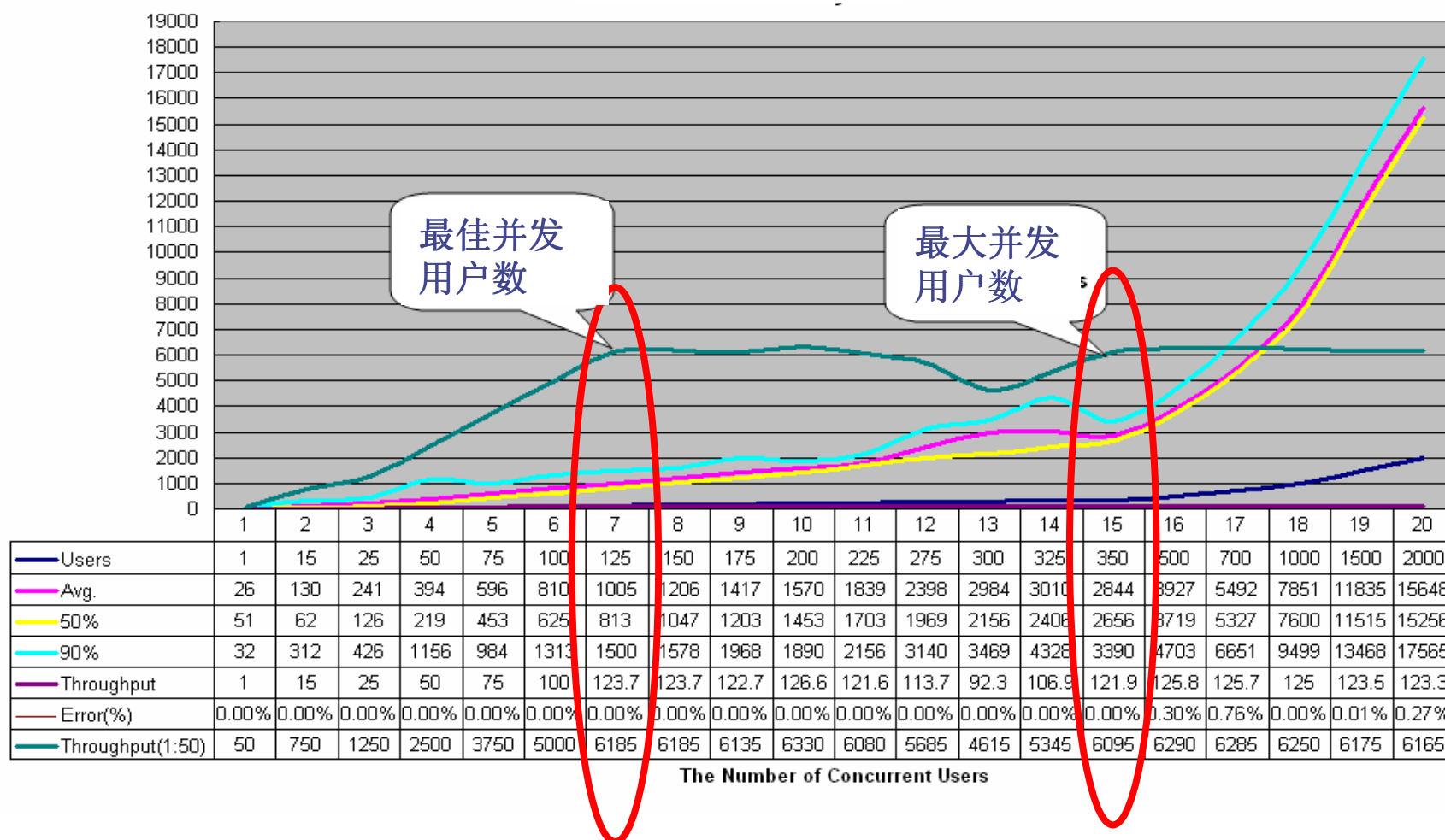
# “理发店”的性能模型





WWW.GZTEST.NET

# 一个实际项目的测试数据





WWW.GZTEST.NET

# 最佳/最大并发用户数的意义

## ◆ 最佳并发用户数

- 应当大于系统的平均负载
- 当并发用户数持续大于该值，可能会出现部分用户请求失败

## ◆ 最大并发用户数

- 应当大于系统的峰值负载
- 当并发用户数大于该值，则必然会有用户请求失败

# 小结

## ◆ 常用的评价系统性能的指标

- 响应时间(**Response Time**)
- 并发用户数(**The Number of Concurrent Users**)
- 吞吐量(**Throughput**) – 每秒交易数(**Transaction per Second**)
- 资源利用率(**Hardware/Software Resource Utilization**)
- 最佳并发用户数(**The Optimum Number of Concurrent Users**)
- 最大并发用户数(**The Maximum Number of Concurrent Users**)

## ◆ 性能指标之间的关系

- ◆ “此消彼长”, “此长彼消”

# 什么是性能测试

- ◆ 观察系统在一个给定的环境和场景中的性能表现是否与预期目标一致，评判系统是否存在性能缺陷，并根据测试结果识别性能瓶颈，改善系统性能的整个过程

# 内容提要

- ◆ 理解“性能”与“性能测试”
- ◆ 性能测试的过程 与 实践
  - 获取有效的性能需求
  - 设计测试场景
  - 构建性能测试环境
  - 开发测试套件
  - 执行测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 优化改善系统性能
- ◆ 相关主题讨论



# 一个例子

- ◆系统总容量达到日委托**6000**万笔，成交**9000**万笔
- ◆系统处理速度每秒**7300**笔，峰值处理能力达到每秒**10 000**笔
- ◆最大交易响应时间**<3**秒
- ◆实际股东帐号数**3000**万

# 什么是“有效的”性能需求

- ◆ 明确的数字，而不是模糊的语句
- ◆ 有凭有据，合理，有意义
- ◆ 相关人员达成一致



WWW.GZTEST.NET

# 如何获得“有效的”性能需求

- ◆ 客户方提出
- ◆ 根据历史数据来分析
- ◆ 参考历史项目的数据
- ◆ 参考其他同行类似项目的数据
- ◆ 参考其他类似行业应用的数据
- ◆ 参考新闻或其他资料中的数据
- ◆ 80/20 原则

易

难

[首页](#) | [计算机书店](#) | [26类主题店](#) | [购物车: 零售|团购](#) | [客户服务](#)

[分类](#) | [排行](#) | [Java](#) | [.NET](#) | [C++](#) | [嵌入式计算机](#) | [程序设计](#) | [软件工程](#) | [操作系统](#) | [数据库](#) | [图形图像](#) | [安全](#)

商品查询:   [组合查询](#) [RSS订阅](#)

[欢迎光临china-pub网上书店](#) | [登录](#) | [注册](#) | [我的china-pub](#)

[SQL Server 2005: 高手修炼指南](#) | [章立民新书推荐](#) | [深入浅出图书系列](#) | [Java心得笔记](#) | [.NET大局观](#) | [电脑报2006年合订本](#) | [O'Reilly新书展台](#) | [Fedora Core 5 Linux](#)

### 本类热销商品

- Perl语言入门(第四版)(...
- 代码大全(第二版)(一...
- 代码大全(第二版)(英...
- Ajax实战(目前Ajax领域...
- JAVA 2核心技术 卷I: 基...
- PHP与MySQL 5程序设计(...
- Programming ASP.NET中...
- C#高级编程(第4版)(...
- Eclipse权威开发指南(...
- Lucene in Action中文版
- 框架设计(第2版): CL...
- Visual C# 2005程序开发...
- JavaScript高级程序设计

### Ruby on Rails: Up and Running (英文影印版)

4小时商品出库



市场价:	¥ 26.00	
4-5星会员:	¥ 18.20 (70折)	返18分
1-3星会员:	¥ 18.98 (73折)	返18分
普通会员:	¥ 19.50 (75折)	返19分

◆ 19个城市货到付款 38家高校免费送货 600城市快递上门 ◆

**【评价】** ★★★★★ [参与评论](#) [查看书评\(共6条\)](#)  
**【原书名】** Ruby on Rails: Up and Running [\[原书信息\]](#)  
**【原出版社】** O'Reilly  
**【作者】** (美) Bruce A. Tate, Curt Hibbs [\[同作者作品\]](#) [\[作译者介绍\]](#)

### 您的购物车

零售购物车里有 1 件商品  
团购购物车里有 0 件商品  
[查看: 零售/团购](#)

### 希望对您有帮助

[您曾浏览过的商品](#) | [查看藏书阁](#)  
 ▶ [买邮电1本书, 6种精美赠品随您拿! 100%中奖](#)  
 ▶ [章立民新书前500册7折](#)  
 ▶ [侯捷 Effective C++ 7折](#)  
 ▶ [china-pub考研图书7折!](#)

### 重点推荐

[http://www.china-pub.com/search/power\\_search/power\\_search.asp?key1=testin](http://www.china-pub.com/search/power_search/power_search.asp?key1=testin)  
<http://www.china-pub.com/computers/common/info.asp?id=32252>  
<http://www.china-pub.com/member/buybook/view.asp?add=1&tid=123830>

### 3.2006数字生活十大可能

<http://www.qnsb.com.cn/gb/node2/node23/node38/userobject1ai27741.html>

- ◆ 其中提到了“截至2005年11月底，杭州数字电视用户数已经突破50万户大关”，以及“一些热门的电视剧的点播量非常高，比如韩国电视剧《大长今》就曾创下了一天十几万次的高点击量。”

### 4. 思华科技再次携手杭州网通 提升宽带增值服务质量

[http://www.onewaveinc.com/cn/events/content.php?id=19&news\\_type=%E5%AA%92%E4%BD%93%E6%8A%A5%E9%81%93&oq=2](http://www.onewaveinc.com/cn/events/content.php?id=19&news_type=%E5%AA%92%E4%BD%93%E6%8A%A5%E9%81%93&oq=2)

- ◆ 杭州网通的服务器每天的点播量评价在5万左右，最高时超过10万人次
- ◆ 系统扩容后流媒体平台将从原来的并发2000用户上升到满足服务全城区3600~4000用户并发点播的业务

### 5.杭州人对电视剧改变“看”法 6万家庭选择互动电视（2005-8-19）

[http://www.jxpd.com.cn/05zjnews/system/2005/08/19/006275320\\_01.shtml](http://www.jxpd.com.cn/05zjnews/system/2005/08/19/006275320_01.shtml)

- ◆ “每集连续剧的点播费用在0.3元—0.5元，点播电影一般为1—3元”
- ◆ “6万户的家庭，平均每天的点播次数在4次左右”
- ◆ “如果按连续剧每集0.5元的费用计算，那么这些家庭相当于每天要花2元钱在点播节目上，一个月下来就是60元的点播消费，大大超过了14元/月的有线电视视听费”。



WWW.GZTEST.NET

# Q & A



WWW.GZTEST.NET

*Have a*

*break*



# 内容提要

- ◆ 理解“性能”与“性能测试”
- ◆ 性能测试的过程 与 实践
  - 获取有效的性能需求
  - 设计测试场景
  - 构建性能测试环境
  - 开发测试套件
  - 执行测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 优化改善系统性能
- ◆ 相关主题讨论



# 为什么要关注测试环境？

- ◆避免不必要的因素成为“瓶颈”
- ◆性能测试是“环境依赖”的测试
- ◆模拟用户环境 vs. 无法模拟的环境



WWW.GZTEST.NET

# 需要确认的项目

## ◆ 硬件型号，配置，数量

- 服务器
- PC
- 其他硬件

## ◆ 软件，版本

- 操作系统
- 数据库
- Web 服务器
- 应用服务器
- 其他软件

## ◆ 网络结构，带宽

## ◆ 历史数据量

- 基础数据
- 业务数据

# 搭建和配置环境

◆ 硬件，软件，网络

◆ 使用“最佳实践”配置

- ◆ 软件版本
- ◆ 数据库配置
- ◆ **JVM**
- ◆ **JDBC** 连接池
- ◆ 线程池
- ◆ 网络带宽
- ◆ ...

◆ 向数据库中注入历史数据

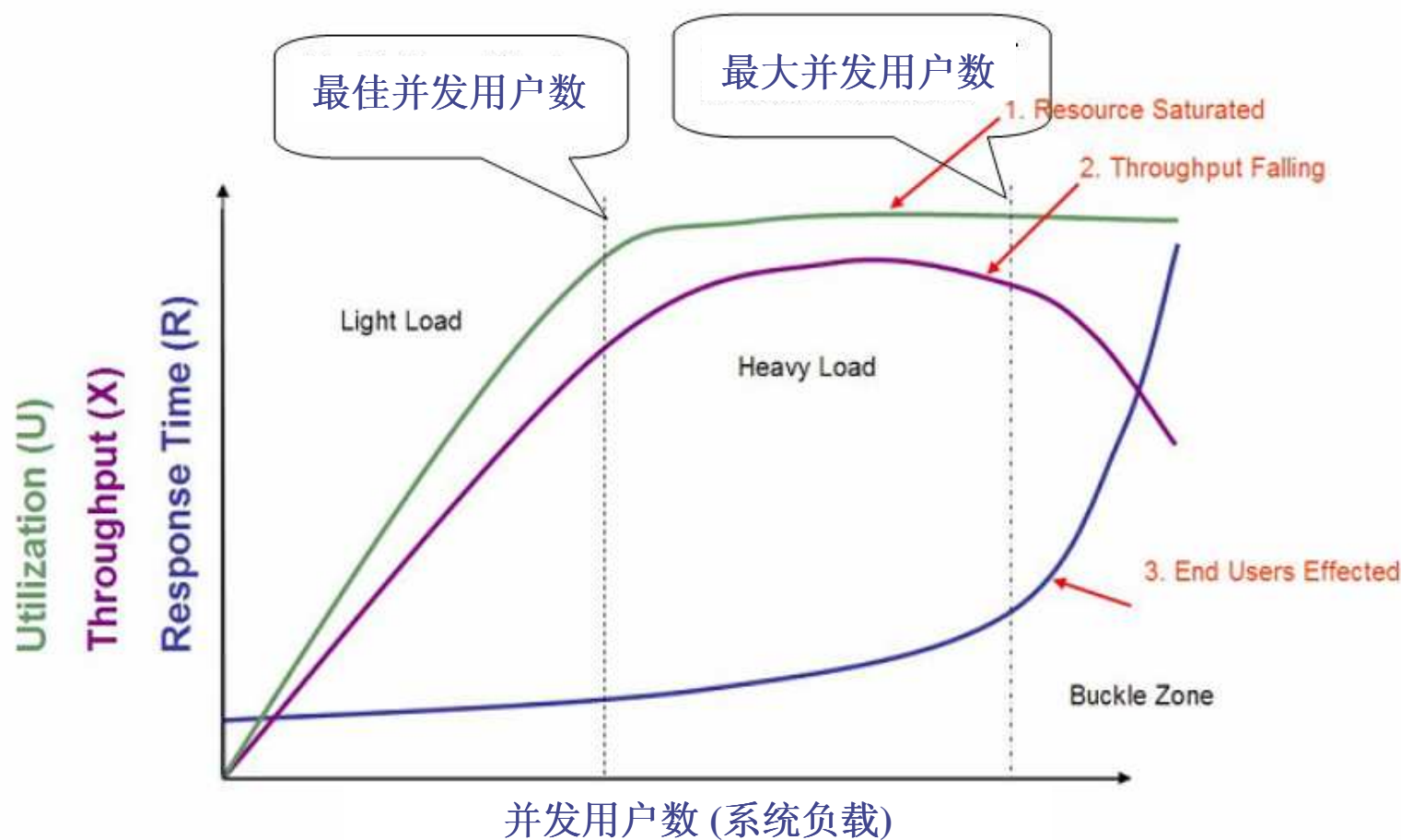
# 内容提要

- ◆ 理解“性能”与“性能测试”
- ◆ 性能测试的过程 与 实践
  - 获取有效的性能需求
  - 设计测试场景
  - 构建性能测试环境
  - 开发测试套件
  - 执行测试
  - 性能测试结果的分析和性能瓶颈的识别
  - 优化改善系统性能
- ◆ 相关主题讨论



WWW.GZTEST.NET

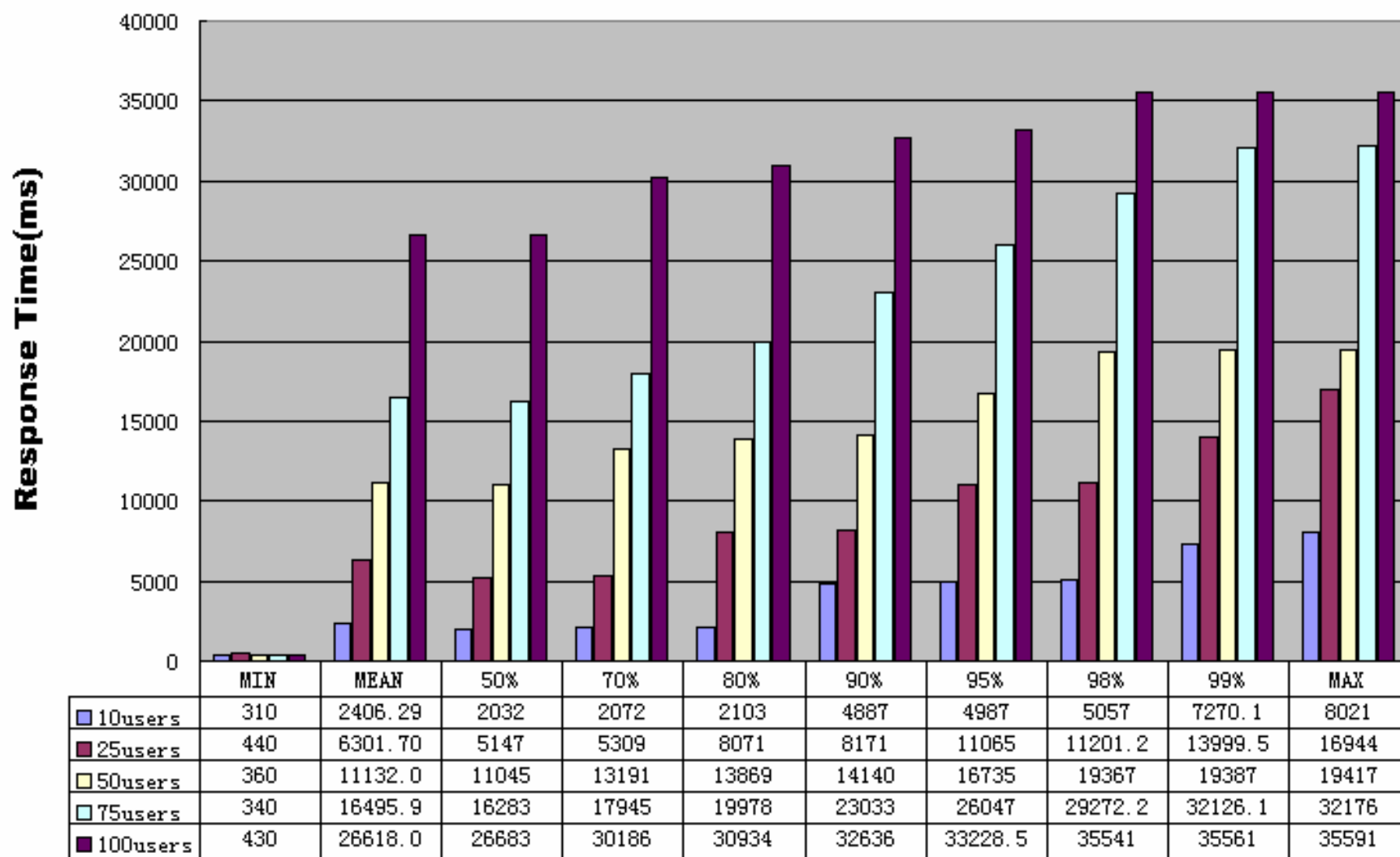
# 最佳/最大并发用户数的分析





WWW.GZTEST.NET

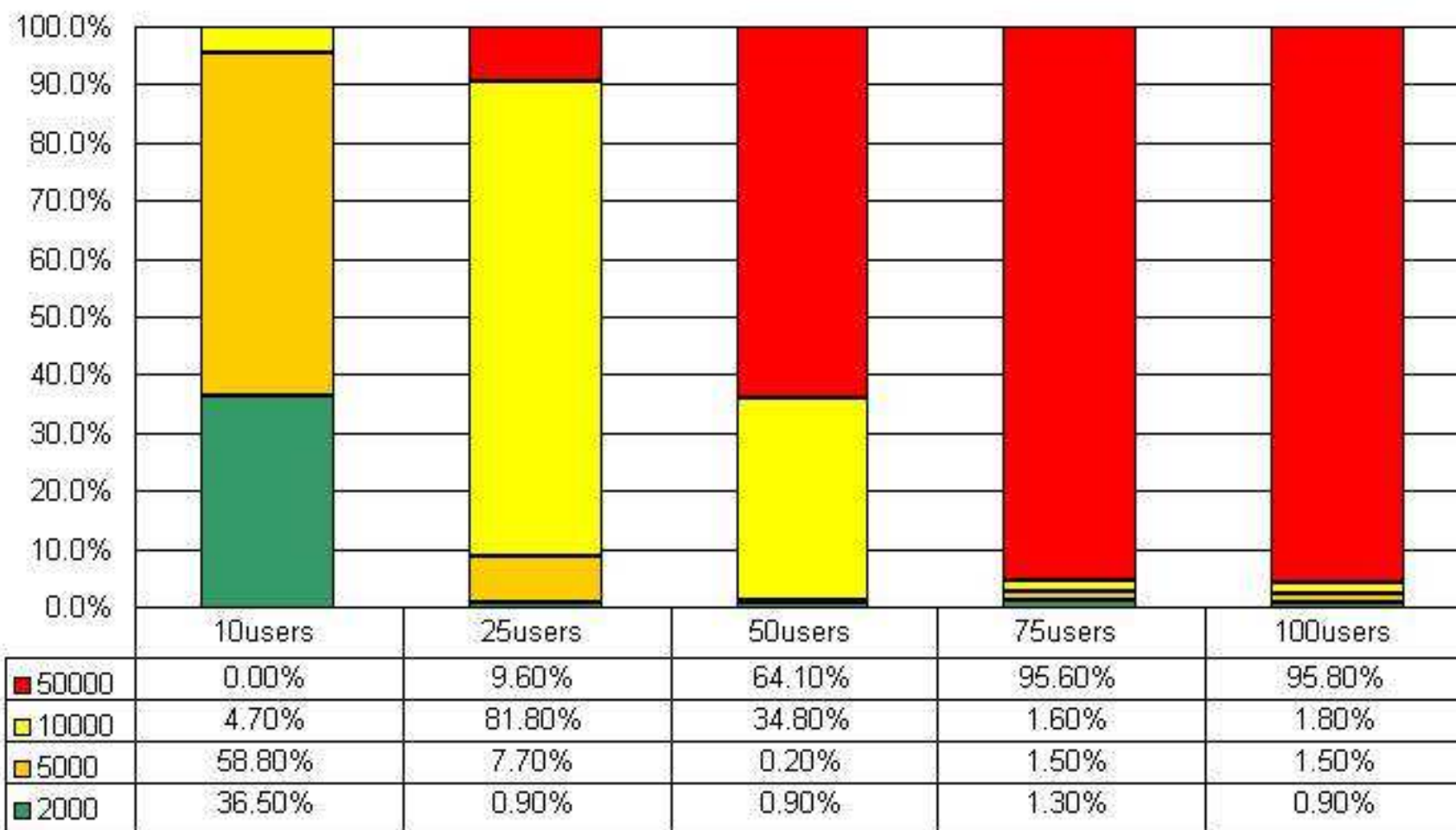
# 响应时间分布情况的分析





WWW.GZTEST.NET

# 响应时间分布情况的分析



Response Time(ms)

# 分析请求失败的原因

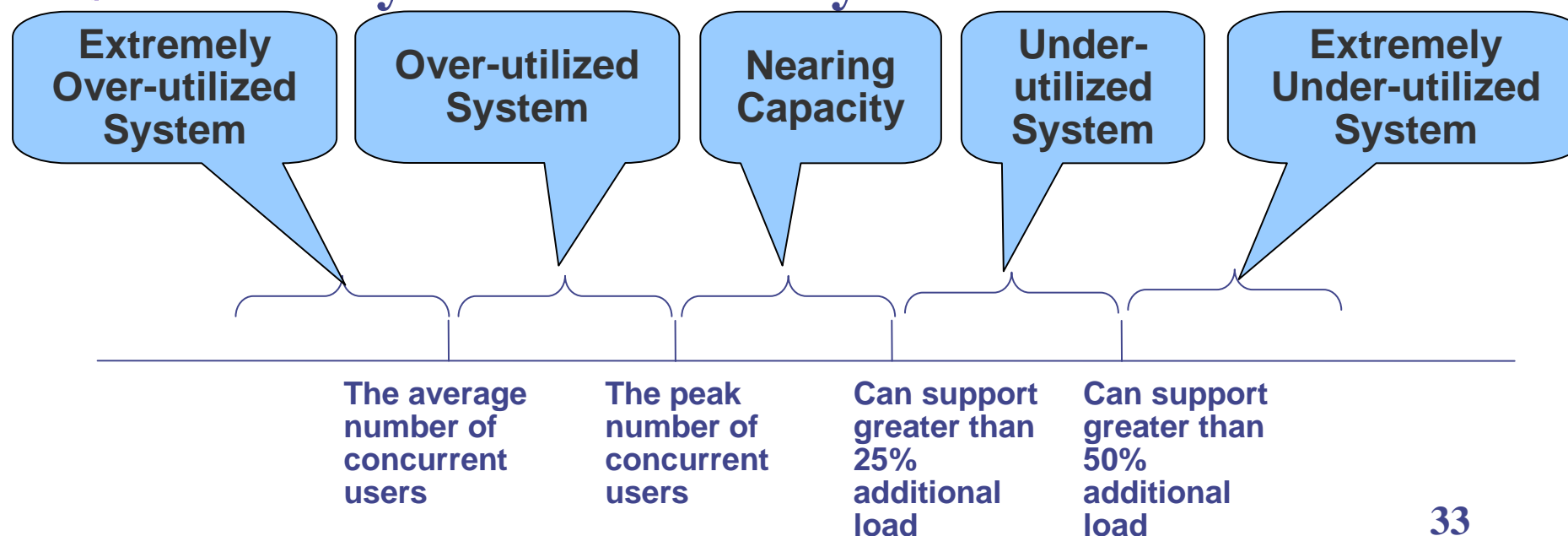
- ◆ 请求超时
- ◆ 连接被拒绝
- ◆ 应用服务器 **error**
- ◆ 数据库 **error**
- ◆ 系统崩溃





# 系统的分类

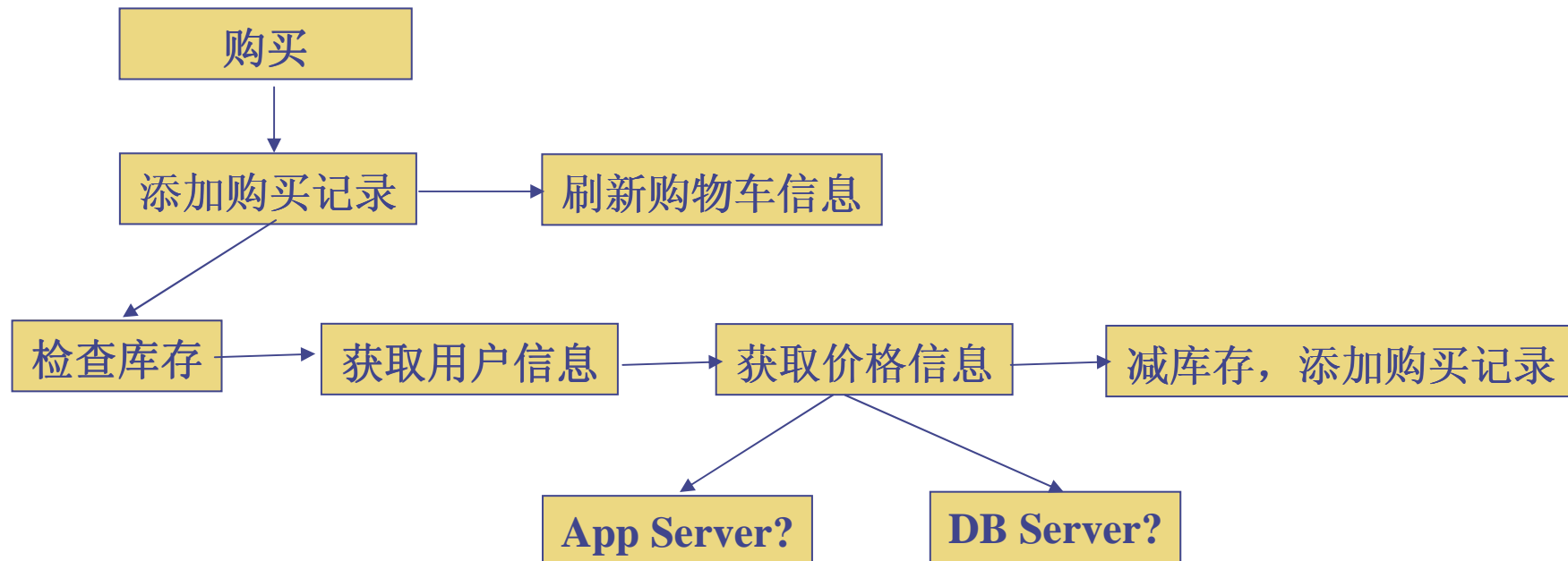
- ◆ **Extremely Under-utilized System**
- ◆ **Under-utilized System**
- ◆ **Nearing Capacity**
- ◆ **Over-utilized System**
- ◆ **Extremely Over-utilized System**



# 识别性能瓶颈的方法

◆ 自下而上的方法

◆ 自上而下的方法





WWW.GZTEST.NET

# Q & A



WWW.GZTEST.NET

# *Thanks!*

