中国开发者软件工程组织

IBM

Cdse

性能测试最佳实践

陈 雷 **Jackei.Chan**

jackeichan@gmail.com

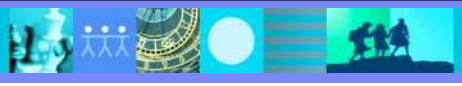
http://jackei.cnblogs.com

Rational. software

中国开发者软件工程组织（CDSE)

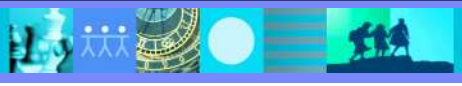
自我介绍

- 广州软件测试行业协会创建人，专家顾问团队负责人
- 《程序员》杂志专栏作者
- 外企资深软件测试工程师
- 多年医疗卫生行业，电信等行业，运营商级系统开发测试经验
- 关注于测试过程改进，性能测试，软件测试自动化，以及开源技术在企业中的应用



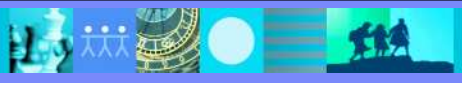
内容提要

- 理解"性能"与"性能测试"
- 性能测试的过程 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 最佳实践
- 相关主题讨论

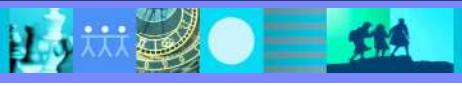


什么是性能测试

- 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

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 最佳实践
- 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)
- 客户(customer)和开发者(developer)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)

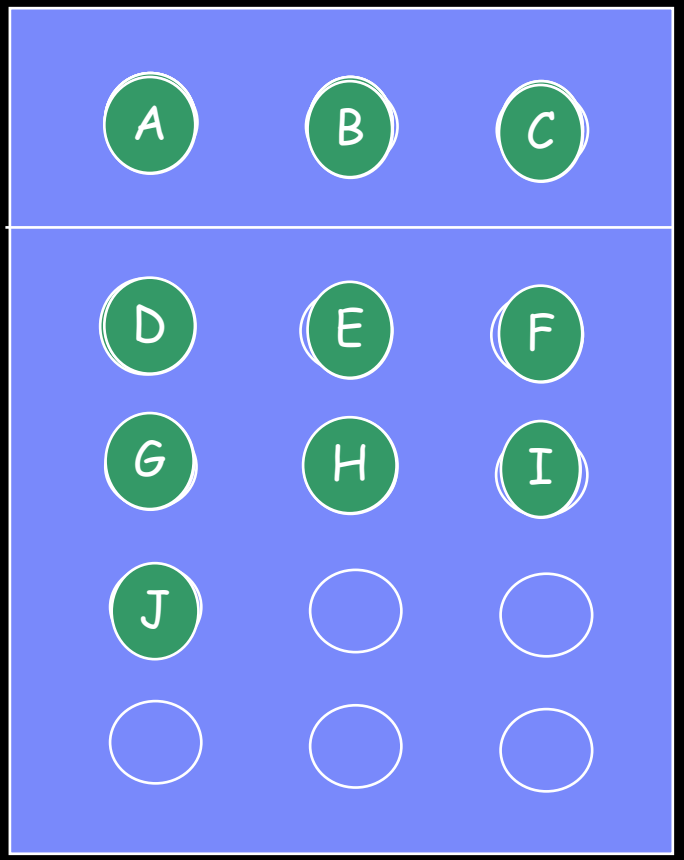
内容提要

- 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 最佳实践
- 相关主题讨论

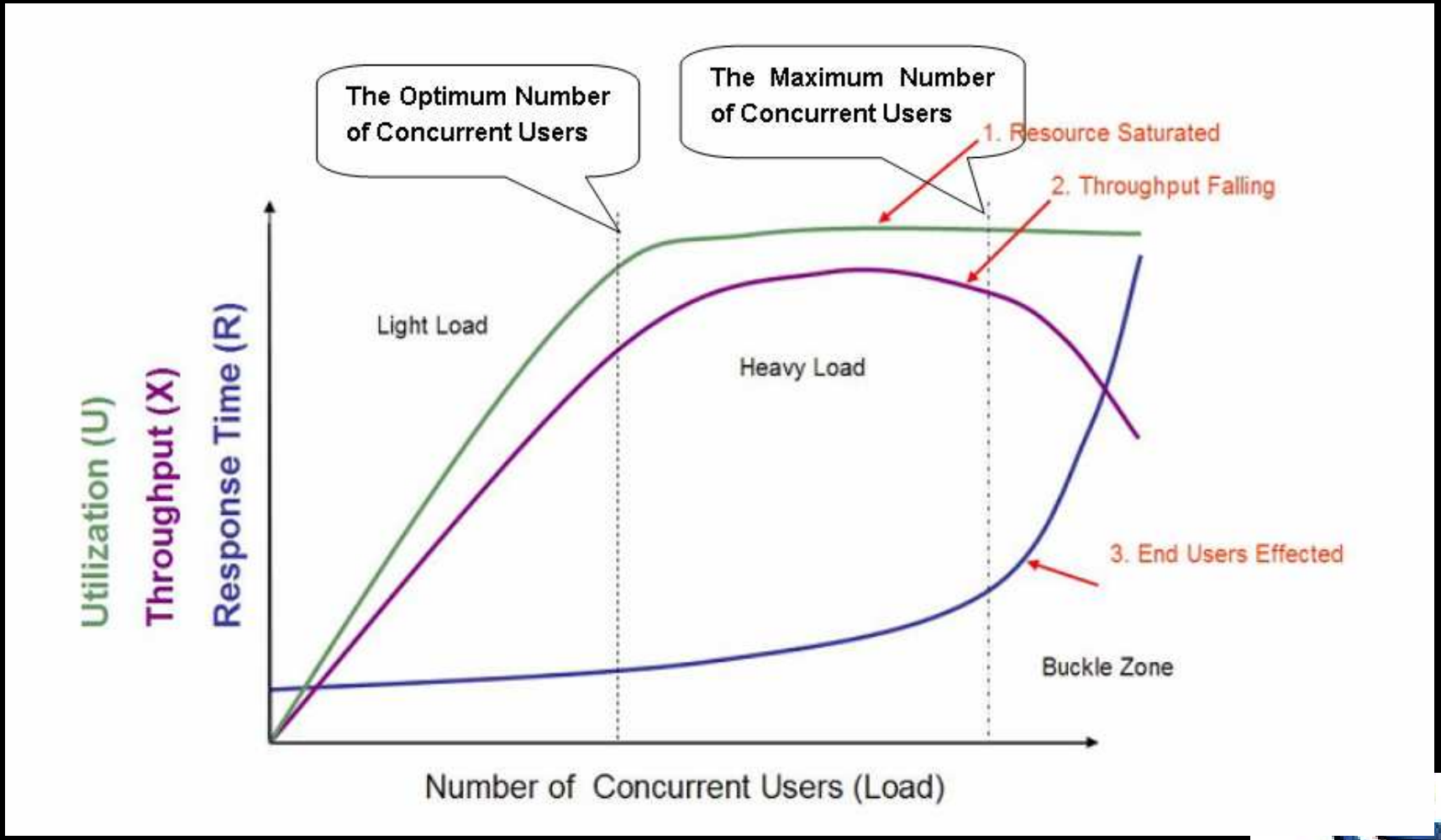

理发师
顾客

理发店模型的 3 个假设

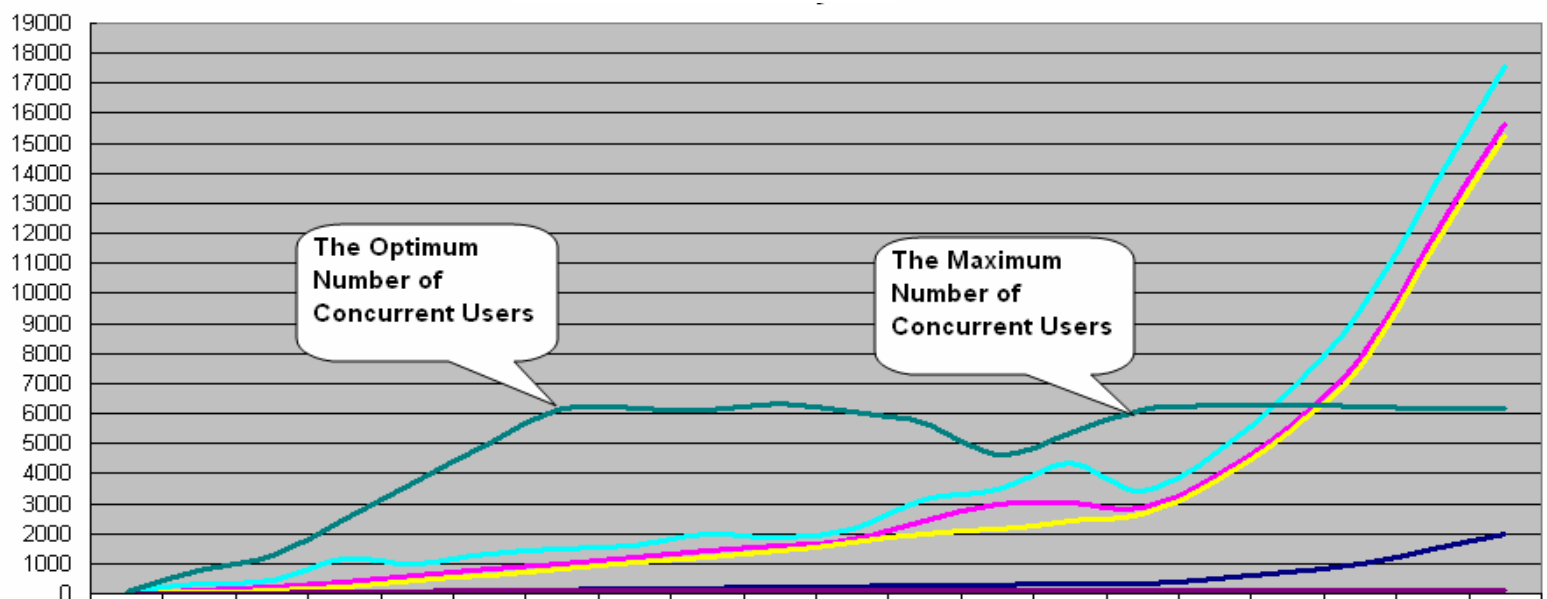
- 理发店中一共有 3 名理发师
- 每位理发师剪一个发的时间都是 1 小时
- 我们顾客们都是很有时间观念的人而且非常挑剔，他们对于每次光顾理发店时所能容忍的等待时间+剪发时间是3小时，而且等待时间越长，顾客的满意度越低。如果3个小时还不能剪完头发，我们的顾客会立马生气的走人



"理发店"的性能模型


The Optimum Number of Concurrent Users
The Maximum Number of Concurrent Users
1. Resource Saturated
2. Throughput Falling
Light Load
Heavy Load
3. End Users Effected
Buckle Zone
Utilization (U)
Throughput (X)
Response Time (R)
Number of Concurrent Users (Load)

一个实际项目的测试数据

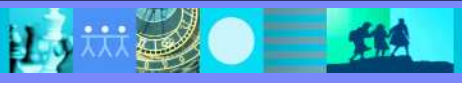


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Users	1	15	25	50	75	100	125	150	175	200	225	275	300	325	350	500	700	1000	1500	2000
Avg.	26	130	241	394	596	810	1005	1206	1417	1570	1839	2398	2984	3010	2844	3927	5492	7851	11835	15648
50%	51	62	126	219	453	625	813	1047	1203	1453	1703	1969	2156	2406	2656	3719	5327	7600	11515	15256
90%	32	312	426	1156	984	1313	1500	1578	1968	1890	2156	3140	3469	4328	3390	4703	6651	9499	13468	17565
Throughput	1	15	25	50	75	100	123.7	123.7	122.7	126.6	121.6	113.7	92.3	106.9	121.9	125.8	125.7	125	123.5	123.3
Error(%)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.30%	0.76%	0.00%	0.01%	0.27%
Throughput(1:50)	50	750	1250	2500	3750	5000	6185	6185	6135	6330	6080	5685	4615	5345	6095	6290	6285	6250	6175	6165

The Number of Concurrent Users

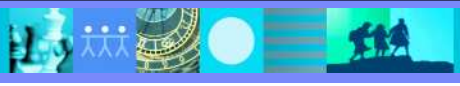
最佳/最大并发用户数的意义

- 最佳并发用户数
  - 应当大于系统的平均负载
  - 是用来对外公布的性能数据
  - 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
- 用于其他测试
  - 可靠性测试（最佳并发用户数）
  - 可伸缩性测试（最佳并发用户数）
  - 可恢复性测试（最佳/最大并发用户数）

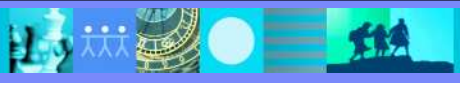


小结

- 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”



Q&A



广州软件测试协会
GUANGZHOU SOFTWARE TEST EGEC

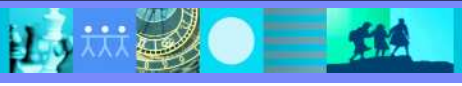
Have a break

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 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论

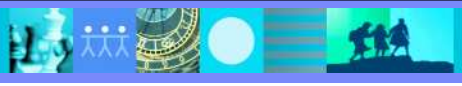


一个例子

- 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
- 最大响应时间小于5秒
- 实际股东帐号数3000万

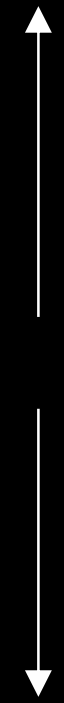
什么是"有效的"性能需求

- 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



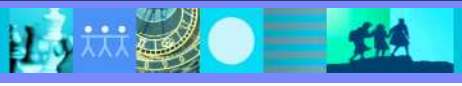
如何获得"有效的"性能需求

- 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



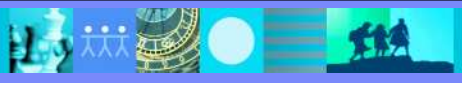
内容提要

- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论



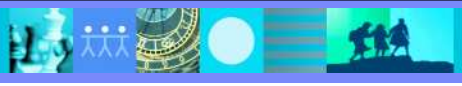
需要确认的项目

- 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



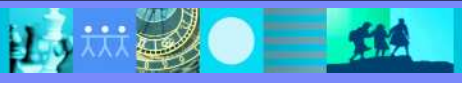
搭建和配置环境

- 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



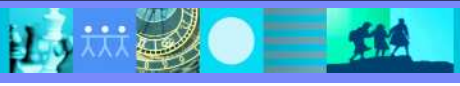
内容提要

- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论



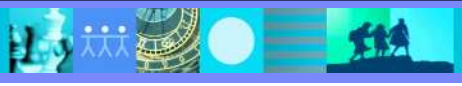
开发和调试脚本

- 录制脚本
- 回放脚本
- 编辑脚本和参数化脚本
- 调试脚本



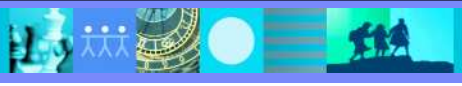
设置测试场景

- 确定加压方式
- 确定需要监控的软硬件资源
- 多业务模块的集成测试策略



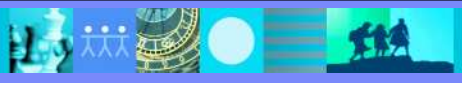
内容提要

- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论



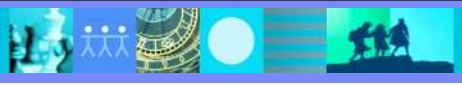
在开始执行测试之前

- 确认脚本已经通过了调试
- 确认测试工具已经正确的配置
- 确认Web/App/DB 服务器是否已经“热身”
- 确认性能监视工具是否可用



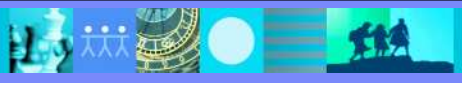
如何执行你的测试

- 逐渐等量的增加并发用户数量
- 使用合适的增量间隔
- 执行足够多的迭代次数或者足够长的时间
- 每次执行后还原测试环境
- 每次执行后立即分析测试结果

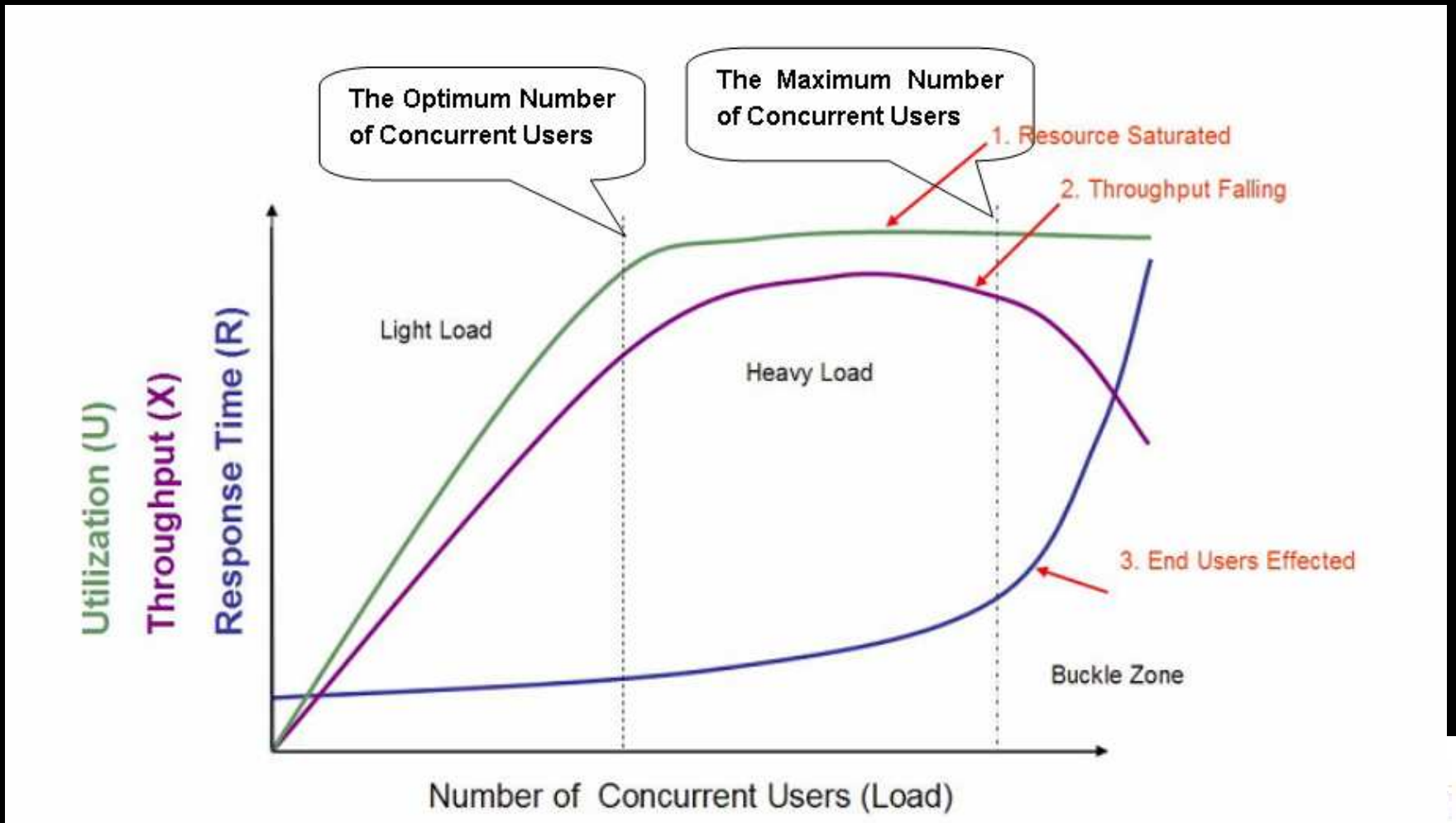


内容提要

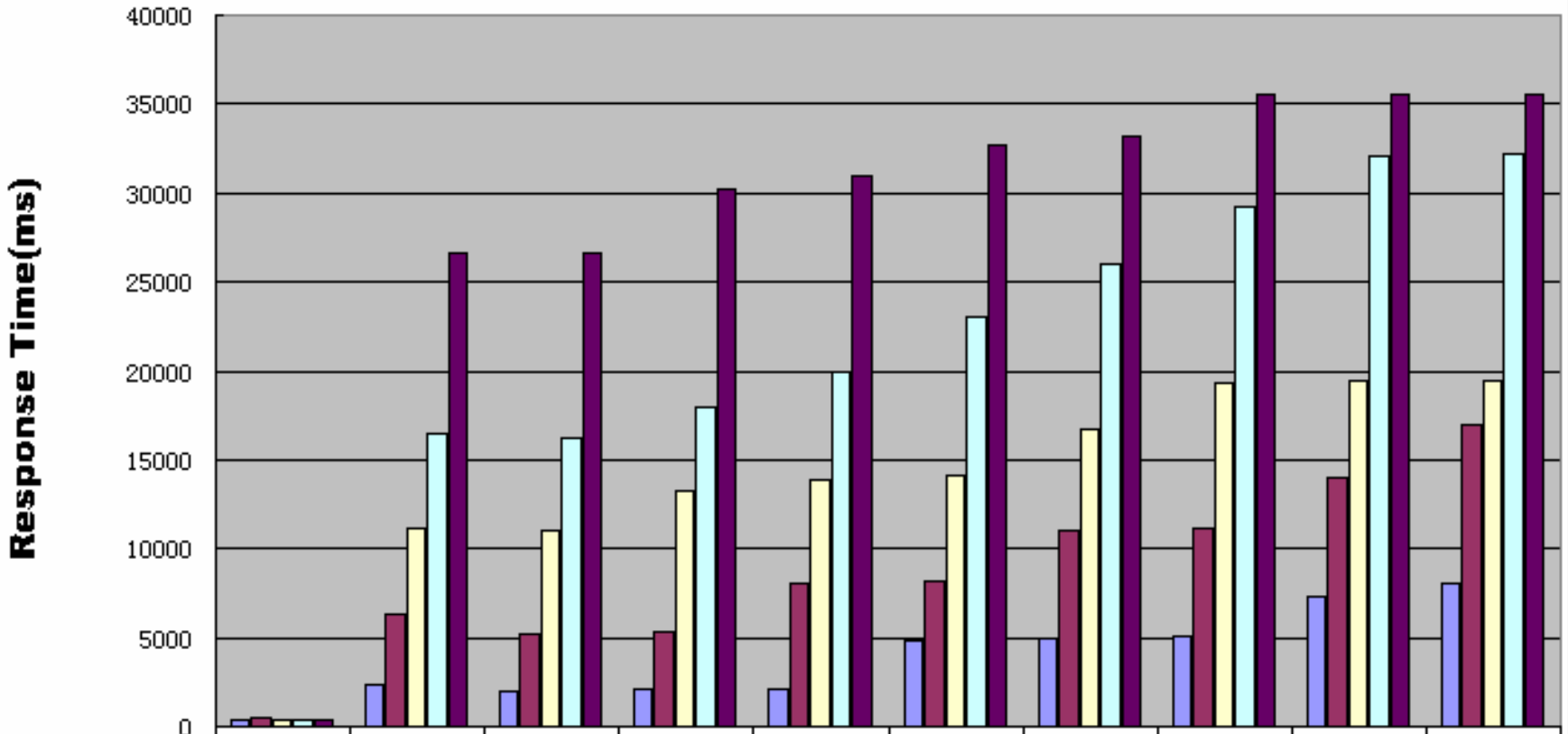
- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论



最佳/最大并发用户数的分析

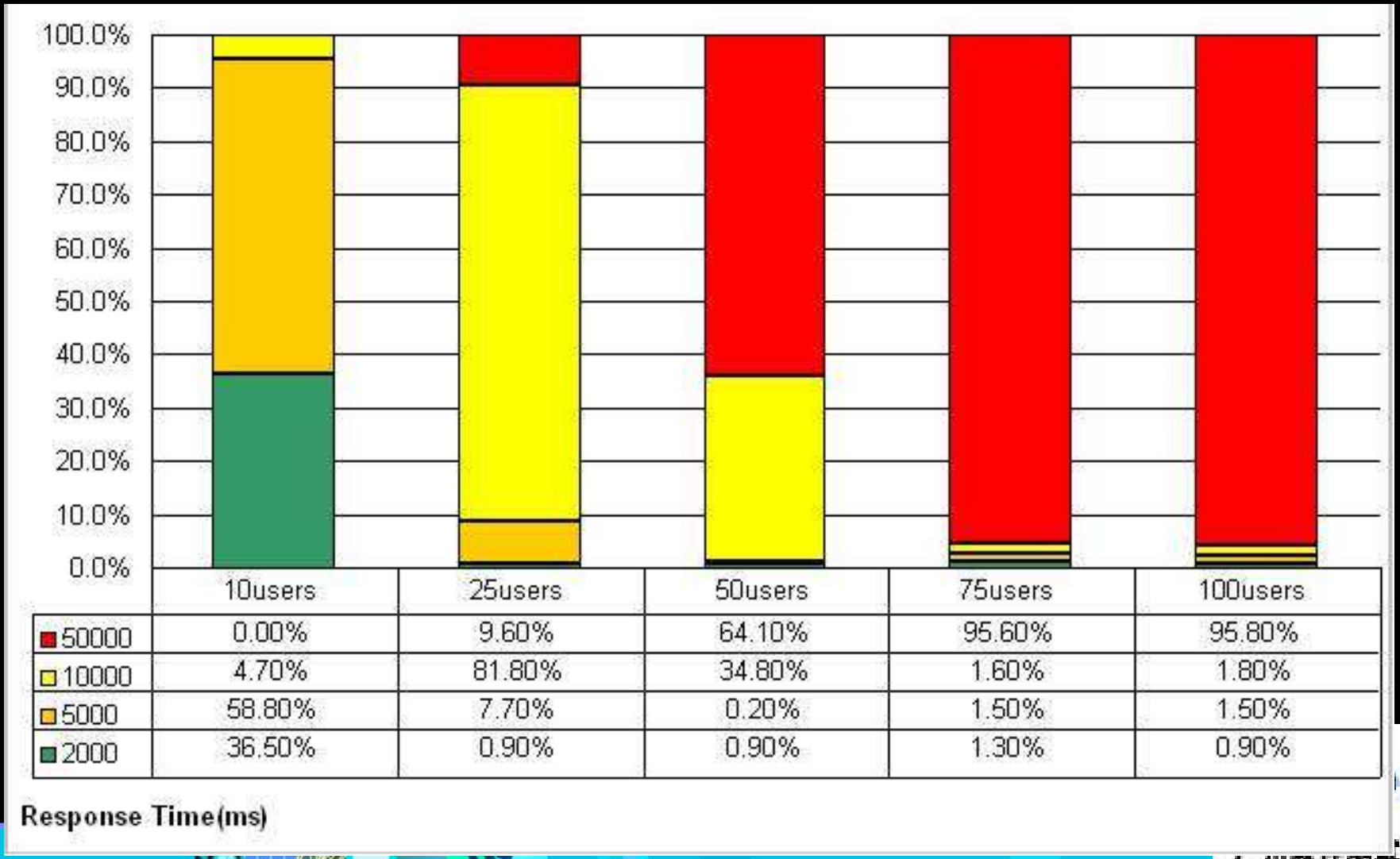


响应时间分布情况的分析



	MIN	MEAN	50%	70%	80%	90%	95%	98%	99%	MAX
10users	310	2406.29	2032	2072	2103	4887	4987	5057	7270.1	8021
25users	440	6301.70	5147	5309	8071	8171	11065	11201.2	13999.5	16944
50users	360	11132.0	11045	13191	13869	14140	16735	19367	19387	19417
75users	340	16495.9	16283	17945	19978	23033	26047	29272.2	32126.1	32176
100users	430	26618.0	26683	30186	30934	32636	33228.5	35541	35561	35591

响应时间分布情况的分析（续）

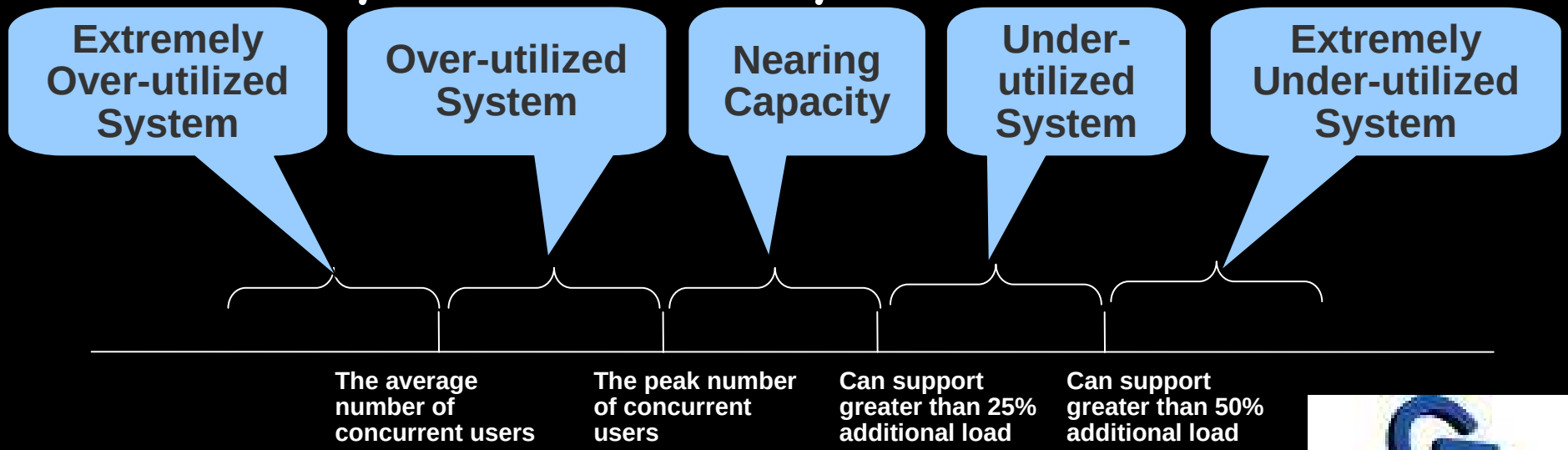


	10users	25users	50users	75users	100users
50000	0.00%	9.60%	64.10%	95.60%	95.80%
10000	4.70%	81.80%	34.80%	1.60%	1.80%
5000	58.80%	7.70%	0.20%	1.50%	1.50%
2000	36.50%	0.90%	0.90%	1.30%	0.90%

Response Time(ms)

系统的分类

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

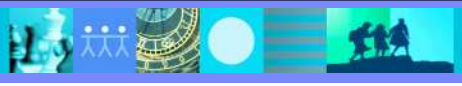


分析请求失败的原因

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

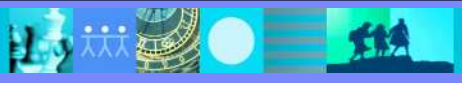
识别性能瓶颈的方法

- 自上而下的方法
- 自下而上的方法



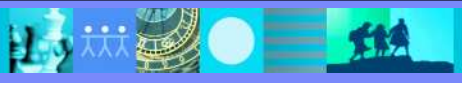
内容提要

- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
  - 获取有效的性能需求
  - 构建性能测试环境
  - 设计和实现性能测试场景
  - 执行性能测试
  - 性能测试结果的分析 与 性能瓶颈的识别
  - 改善系统的性能
- 相关主题讨论



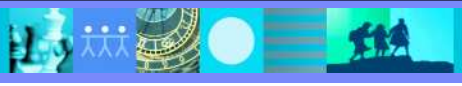
性能调优

- 只作你能做的
- 每次只作一处改动
- 改动前备份相关的内容
- 记录每次的改动
- 依靠团队的力量

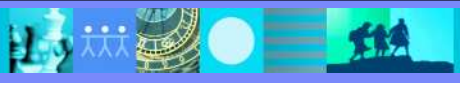


增强性能的可扩展性

- 可扩展性技术
- 性能可扩展性与系统复杂度的平衡



Q&A

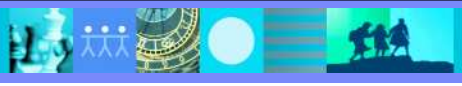


广州软件测试协会
GUANGZHOU SOFTWARE TEST EGEC

Have a break

内容提要

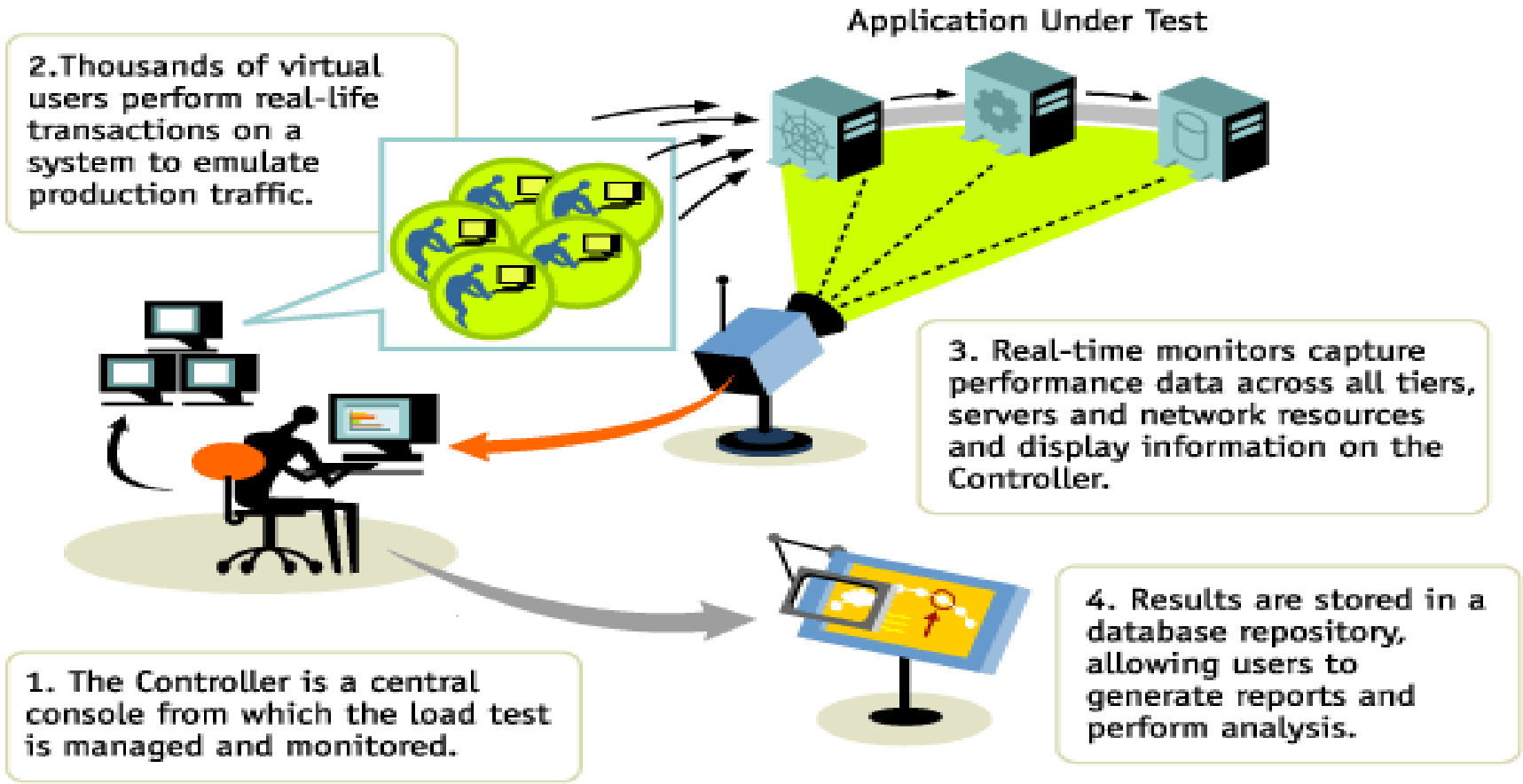
- 理解"性能"与"性能测试"
- 性能测试的过程 与 最佳实践
- 相关主题讨论
  - 编写性能测试报告的要点
  - 性能测试工具的工作原理
  - 如何选择性能测试工具
  - 无处不在的性能测试



编写性能测试报告的要点

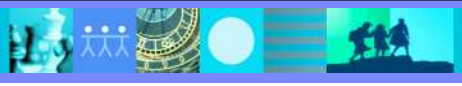
- 系统是否满足性能需求
- 系统所属的分类
- 汇总测试结果
- 你的建议？
- 使用数据表和图表

性能测试工具的工作原理



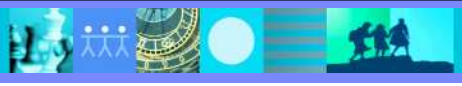
如何选择性能测试工具

- 功能性
  - 协议
  - 可编程性
  - 多种负载模式
  - 性能监视器工具
  - 分析工具
  - 参数化
  - IP欺骗
  - 关联



如何选择性能测试工具（续）

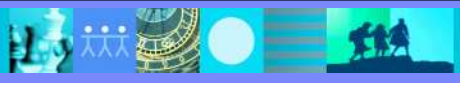
- 成本
  - License
  - 培训
  - 技术支持
  - 升级
- 常见的性能测试工具
  - 商业工具：LoadRunner
  - 开源/免费工具：JMeter，ab，OpenSTA



无处不在的性能测试

- 需求阶段的性能测试
- 系统分析设计阶段的性能测试
- 编码阶段的性能测试
- 维护阶段的性能测试
- 可靠性测试(Reliability Testing)
- 可伸缩性测试(Scalability Testing)
- 可恢复性测试(Recoverability Testing)

Q&A



广州软件测试协会
GUANGZHOU SOFTWARE TEST EGEC

Thanks!