

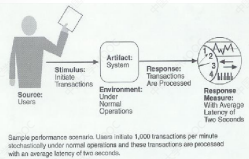
## 性能

- 企业应用系统中有很多性能糟糕的例子
- 性能：
  - 单位时间内完成的工作量的度量
  - 必须满足的截止期限
- 企业应用通常有严格的性能需求，比如：
  - 每秒1000个事务
  - 每个请求的平均延迟时间为3秒
- 将关键操作本地化并尽量减少通信；使用大粒度的组件。

## 通用性能场景表

场景组成	可能的值
激励源	多个独立激励源之一，可能来自系统内部
激励	周期性事件；偶然事件；随机事件
工件	系统
环境	正常模式；过载模式
响应	处理刺激；改变服务级别
响应度量	延迟；截止期限；吞吐量；缺失率；数据丢失

## 性能场景例子



## 性能 - 吞吐量

- 应用程序在单位时间内必须执行的工作量
- 每秒处理的事务数量
- 每分钟处理的消息数
- 需要的吞吐量：
  - 平均值是多少？
  - 负载峰值时是多少？

## 性能- 响应时间

- 应用程序处理一个请求的延迟时间。
- 通常以（毫）秒为单位
- 需要的响应时间：
  - 最大延迟是多少？
  - 平均延迟是多少？

## 性能 - 截止期限

- 必须在某个特定时间之前完成
- 工资发放系统必须在凌晨两点前结束工作，以便电子转账信息被发送到银行
- 每周审计必须在周一早上六点完成以便管理层可以使用数据
- 截止期限通常与IT系统中的批处理相联系。

## 可扩展性

- 可扩展性描述了系统、网络或进程以用户可以接受的方式处理工作量的增长或能更容易地扩张。
- IT系统中四个常见的可扩展性问题:
  - 请求负载
  - 连接数
  - 数据量
  - 部署节点数

## 可扩展性——请求负载

- 当100tps应用程序的并发请求负载增加时，应用程序如何反应？
  - 从每秒100个请求增加到每秒1000个请求？
- 无需额外硬件容量的理想解决方案:
  - 负载增加时，吞吐量保持常量（如100tps），每个请求的响应时间仅线性增长（如10秒）。

## 可扩展性 - 连接数

- » 一个程序的并发连接数增加时会发生什么？
  - 每个连接都消耗一个资源会怎样？
  - 超过最大连接数会怎样？

## 可扩展性 - 数据大小

- » 系统处理的数据量增加时会表现如何？
  - 聊天应用程序的平均消息量加倍？
  - 数据库中数据量从1百万行增加到2千万行？
  - 图像分析算法处理100MB的图片而不是仅仅1MB？
- » 程序或算法能否处理不断增长的数据量？

## 可扩展性- 部署节点数

- » 当部署节点数增加时，安装、部署的成本如何？
  - 安装新的用户机？
  - 安装新的服务器？
- » 解决方案通常为自动下载/安装
  - 譬如从互联网下载程序

## 可扩展性

- » 可扩展性经常被忽视
  - 不是应用崩溃的主要原因
  - 难以预测
  - 难以测试/验证
  - 主要依赖验证过的设计和技术