COMP3027J 课程
软件架构
可用性和其策略

邓永健

北京工业大学计算机学院

数据挖掘与安全实验室（DMS 实验室）

COMP3027J 课程
软件架构
可用性和其策略

邓永健

北京工业大学计算机学院

数据挖掘与安全实验室（DMS 实验室）

大纲

1. 质量属性的含义

2. 质量属性场景

3 . 可用性的含义

4. 提升可用性的策略

大纲

1. 质量属性的含义

2. 质量属性场景

3 . 可用性的含义

4. 提升可用性的策略

质量属性（QA）

这是一个非功能性需求，并非由功能决定。

要实现功能特性，系统中的每个部分（模块）都必须被赋予正确的职责、正确的资源以及正确的调度顺序。

先实现功能，然后再讨论质量属性

质量属性（QA）

功能需求
对战；对阵；与.....对抗
非功能性需求

质量属性（QA）

不同的软件项目侧重于不同的质量属性。

影响建筑风格的选择

质量属性可能会相互制约。

优先考虑关键的质量属性

质量属性（QA）

它必须将设计、实施和部署这三个方面结合起来，以满足要求。

不可或缺的

在架构层面，有必要考虑质量属性的实现。

质量属性

常见质量属性

- 可用性。
- 可修改性。
- 性能。
- 安全。
- 可测试性。
- 易用性。

大纲

1. 质量属性的含义

2. 质量属性场景

3 . 可用性的含义

4. 提高可用性的策略

质量属性场景

仅仅指出一个质量属性的名称太过模糊。

客户希望“系统运行速度快”

- 多快算快?
- 系统中的哪些服务应优先考虑以确保速度?

客户想要“高系统安全性”

- 多安全才算安全?

主要威胁有哪些?

质量属性场景

描述该系统如何对刺激作出反应

质量属性场景的六个组成部分

刺激源（刺激的来源/源头）：谁引发了刺激

刺激：影响系统的某种情况

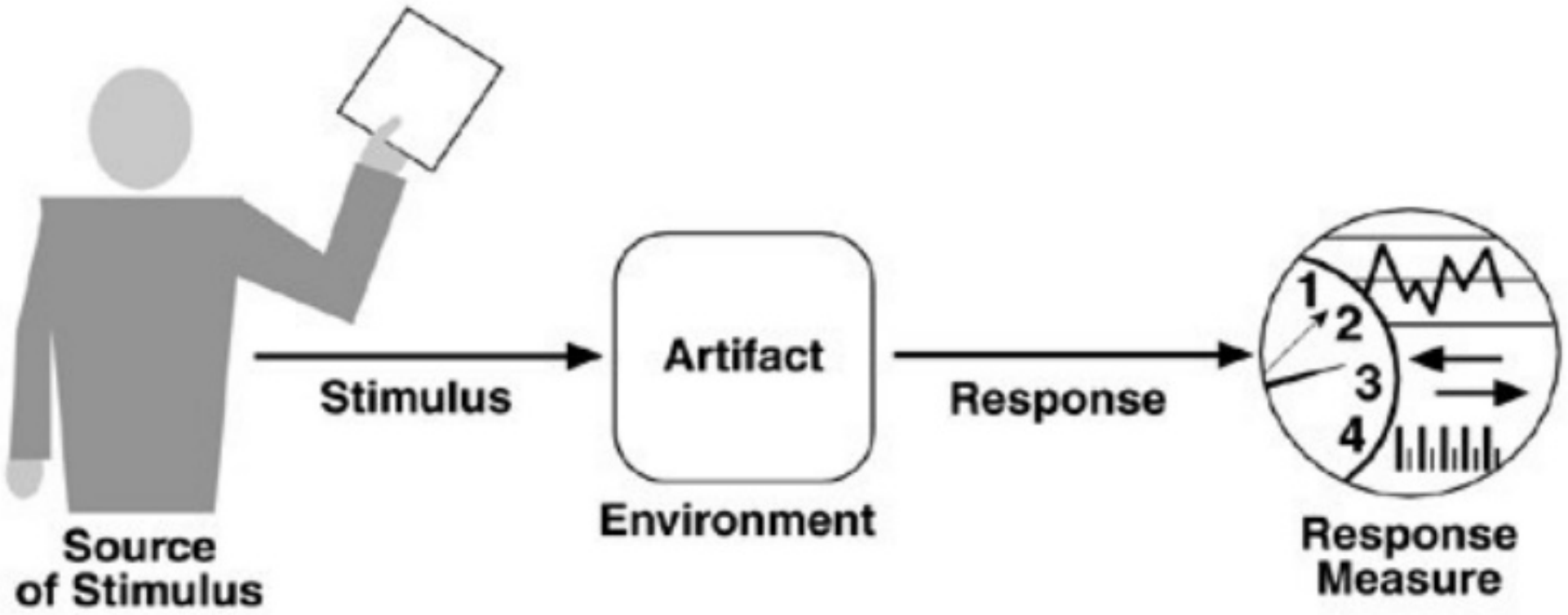
制品（受影响的系统部分）：受系统影响的部分

环境：刺激发生时系统的状态

- 响应（Response）：刺激所引发的结果

响应衡量指标（Response Measure）：如何评估响应

质量属性场景示例


Source
of Stimulus
Stimulus
Artifact
Environment
Response
1
2
3
4
Response
Measure

大纲

1. 质量属性的含义

2. 质量属性场景

3. 可用性的含义

4. 提升可用性的策略

可用性含义

定义

当用户使用系统时，系统可用的概率

计算中不包含提前确定的维护停机时间。

可用性含义

担忧

是否发生了故障（无法提供正常服务，被外界发现）

故障的后果

可用性含义

从 99.9% 的可用性提升到 99.999% 的可用性，这意味着什么?

指标

可用（或停机）时间百分比

修复故障所需时间

平均故障间隔时间

- ……

可用性场景的含义

刺激源

- 失败的迹象（来自内部或外部的因素）

刺激

- 系统故障

系统崩溃（反复出现故障）

结果未能按时给出（过早或过晚）

返回错误结果

可用性场景的意义

人工制品

计算或存储或网络传输

环境

正常状态或“亚健康”状态

可用性场景的意义

反应

记录日志（故障报告）并将其发送回制造商。

通知管理员或其他系统

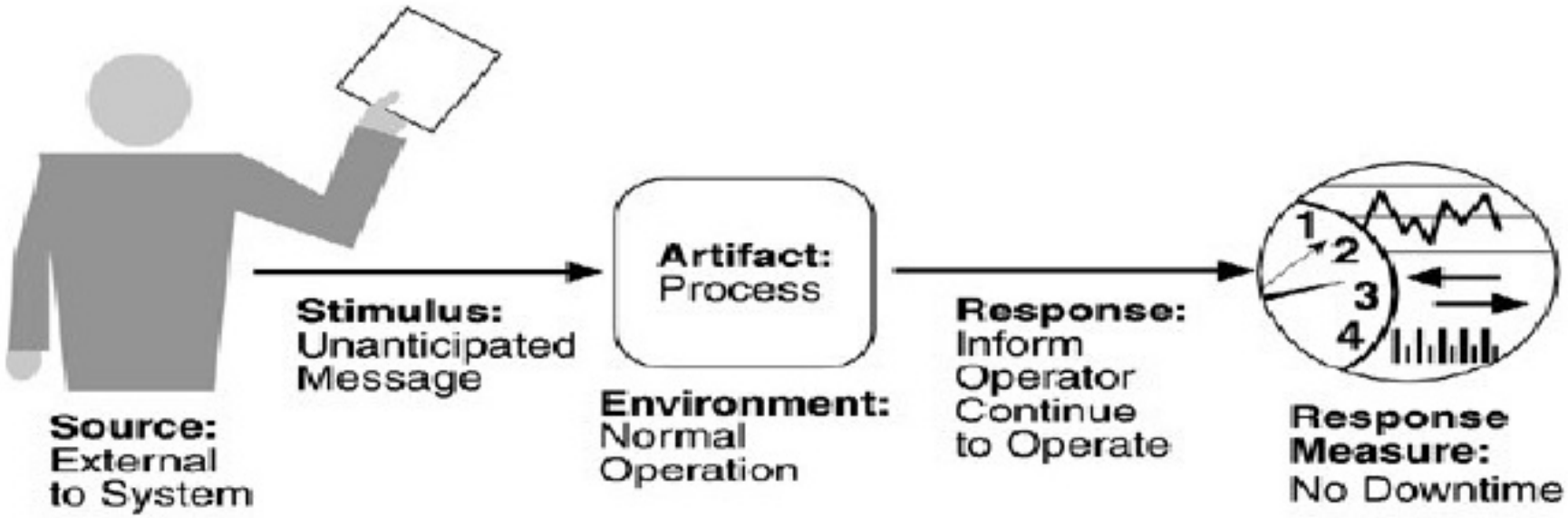
关闭系统；系统在维护期间不可用

应对措施

故障发生前的百分比时间、修复故障所需时间

- … …

可用性的含义——场景示例


Source:
External
to System
Stimulus:
Unanticipated
Message
Artifact:
Process
Environment:
Normal
Operation
Response:
Inform
Operator
Continue
to Operate
1
2
3
4
Response
Measure:
No Downtime

大纲

1. 质量属性的含义

2. 质量属性场景

3. 可用性的含义

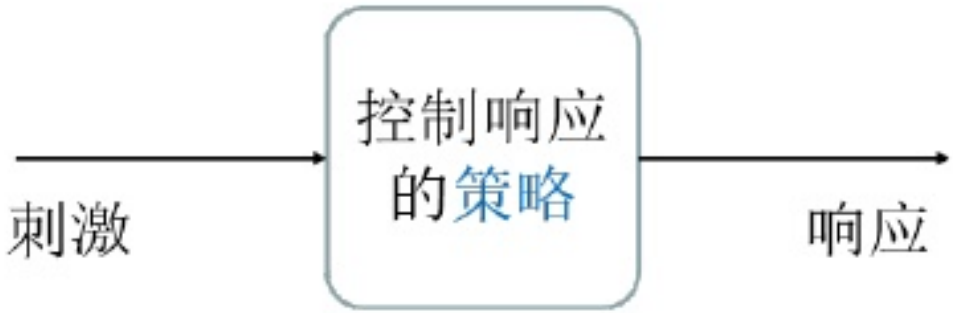
4. 提高可用性的策略

提高可用性的策略 - 定义

战术

具体的设计意味着要满足特定的质量属性。

- 是建筑风格的基本单位



提高可用性的策略 - 定义

目标

降低故障的影响

1. **故障检测**

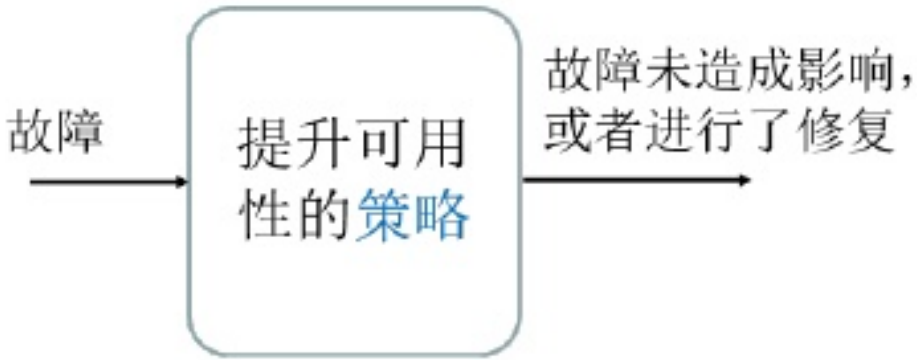
   如何尽早发现故障

2. **故障恢复**

   如何恢复正确结果

3. **故障避免**

   如何主动减少故障的发生



提高可用性的策略——故障检测

“ping/回显”

监控组件会定时向被监控组件发送探测消息，并根据收到的回显消息（是否收到、延迟情况）做出响应。

```
U:\>ping www.baidu.com -t

Pinging www.a.shifen.com [115.239.210.26] with 32 bytes of dat

Reply from 115.239.210.26: bytes=32 time=32ms TTL=53
Reply from 115.239.210.26: bytes=32 time=35ms TTL=53
Reply from 115.239.210.26: bytes=32 time=29ms TTL=53
Reply from 115.239.210.26: bytes=32 time=31ms TTL=53
Request timed out.
Request timed out.
Reply from 115.239.210.26: bytes=32 time=26ms TTL=53
```

提高可用性的策略——故障检测

心跳

被监控组件会定期向监控组件发送心跳消息。

节点之间会持续发送周期性心跳信号以检测每个节点的状态。如果连续未收到的心跳信号数量达到一定数值，则认为相应系统已出现故障。

提高可用性的策略——故障检测

异常

- 抛出 + 捕获 + 处理
- 需要编程语言支持

```
log4j:WARN No appenders could be found for logger (org.hibernate.cfg.Environment).
log4j:WARN Please initialize the log4j system properly.
Exception in thread "main" org.hibernate.HibernateException: /hibernate.cfg.xml not found
        at org.hibernate.util.ConfigHelper.getResourceAsStream(ConfigHelper.java:170)
        at org.hibernate.cfg.Configuration.getConfigurationInputStream(Configuration.java:1453)
        at org.hibernate.cfg.Configuration.configure(Configuration.java:1475)
        at org.hibernate.cfg.Configuration.configure(Configuration.java:1462)
        at org.test.HibernateTest.getCurrentSession(HibernateTest.java:31)
        at org.test.HibernateTest.main(HibernateTest.java:17)
```

提高可用性的策略——故障恢复

投票

- 多个冗余组件，使用统一或不同的算法来完成相同的任务。如果计算结果不同，则少数服从多数。



提高可用性的策略——故障恢复

投票

为降低同时出现故障的概率，可由不同的开发团队在不同的软件和硬件平台上开发多个组件。

提高可用性的策略——故障恢复

主动冗余

服务器 A 和 B 完成相同的计算（A 和 B 的状态始终保持一致），通常只采用 A 计算得出的结果。

当 A 出现故障时，系统能够极快地切换到 B 。

被动冗余

服务器 A 完成操作后会在一定时间内将自身状态通知给 B，然后 B 再将自身状态更新为 A 的状态。

当 A 失效时，您首先需要确认 B 的状态是最新的。

再次上线前，您需要重新同步状态。

提高可用性的策略——故障恢复

内部测试（Alpha 版）

开发人员修复漏洞并在内部进行测试，确认无误后再发布补丁。

检查点/回滚

定期保存，以便轻松恢复

提高可用性的策略——故障避免

服务离线

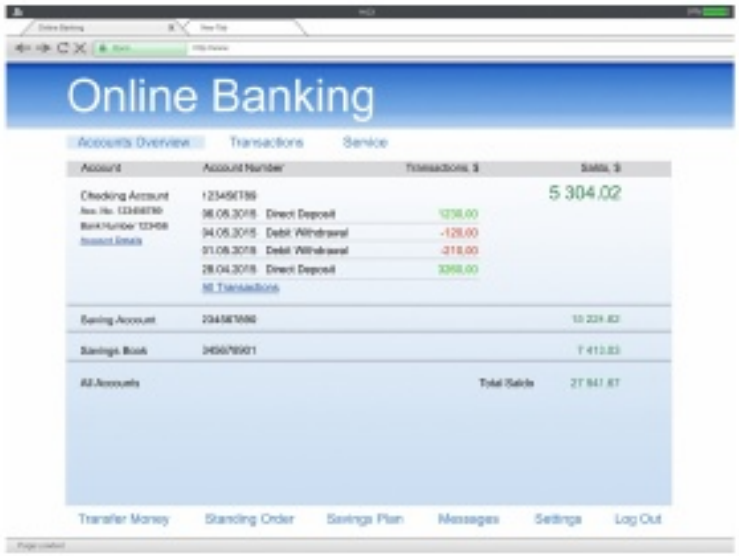
如果您确切知道即将遭受毁灭性的攻击，那么最好主动将服务下线。

提高可用性的策略——故障避免

事务

- 多项操作必须完成

- 案例：A 向 B 转账包含 2 个操作（减少 A 的账户余额和增加 B 的账户余额）。如果任何一个操作失败，另一个操作也必须取消。

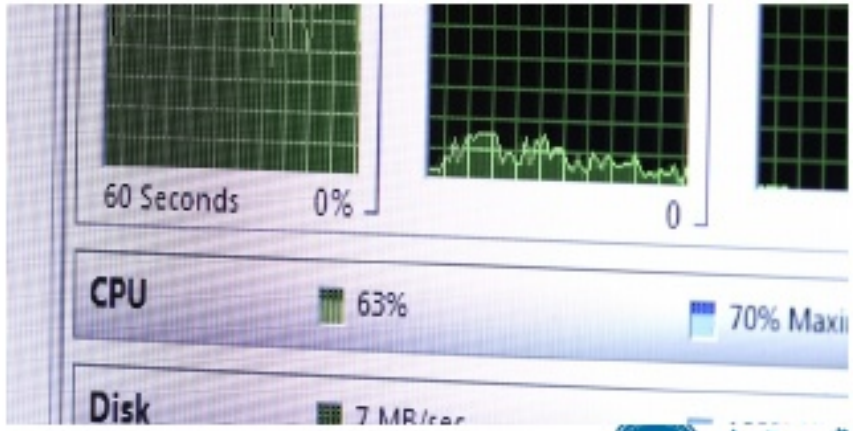


提高可用性的策略

- 故障避免

过程监控

例如，Windows 任务管理器



问答可用性 - 概要

可用性问题

- 故障

提高可用性的策略

故障检测

故障恢复

- 故障避免

谢谢你!