utility tree, sensitivity points and trade-off points

1.Utility tree质量属性效用树

|  |  |  |
| --- | --- | --- |
| 质量属性 | 属性精化  (Attribute Refinement) | ASR  (QA scenarios) |
| 安全性 | 机密性(confidentiality) | 身份验证通过的用户可以进行购票等操作，但是系统应当阻止外部恶意软件伪装合法用户进行的操作，并将伪装用户账号冻结**（H, M）** |
| 完整性(integrity) | 系统在对用户身份进行验证的完成比率不小于99.999%**（L, L）** |
| 应对恶意攻击 | 系统可以检测到外部软件的恶意攻击，同时在10s内采取相应保护措施，并自动上报攻击信息**（M, H）** |
| 可用性 | 错误修复 | 系统自身运行时出现错误，快速修复错误的时间不超过2s，错误修复成功率不低于90%**（H, H）** |
|  | 无停机时间 | 系统可以向用户提供7\*24\*365标准的购票服务**（M, H）** |
| 可延展性 | 硬件升级 | 系统服务器进行硬件上的升级，硬件升级队伍可以在一个工作日内完成任务，升级所影响的代码量不应超过2%**（L, M）** |
| 互操作性 | 与外部系统交互 | 系统调用银行系统提供的服务完成购票支付的操作，交互请求成功率大于99.999%**（H, H）** |
| 易用性 | 新用户学习 | 一个从未使用过系统的用户，可以在系统的指导下，在15分钟内成功完成一次购票操作**（M, M）** |
|  | 正常操作 | 一个使用过系统数次的用户可以熟练进行查票、购票的操作，在整个过程中不需要系统给出操作提示**（H, M）** |
| 性能 | 用户请求的响应时间 | 在系统正常运行的情况下，当用户请求查询余票情况时，系统应当在2s之内给出查询结果**（H, M）** |
|  | 用户请求的响应时间 | 在系统满负载运行的情况下，当用户请求查询余票情况时，系统应当在5s之内给出查询结果**（H, H）** |
|  | 吞吐量 | 在满负载时，系统应当可以在一秒之内处理完10000次查询与5000次更新操作**（H, M）** |

2.敏感点与权衡点列表

敏感点

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 决策 | 质量属性 | 分析 |
| S1 | broker被动冗余 | 可用性 | 能够有效处理错误，但被动冗余的broker作用不完全相同，需要额外信息才能恢复主broker，占用一定资源，Broker的数目会影响系统可用性 |
| S2 | 限制操作时间 | 性能 | 控制一项请求的操作时间，保证处理资源的合理分配，但是检测操作时间本身消耗资源，过高的检测频率会影响处理性能 |
| S3 | 引入并行 | 性能 | 并行处理请求可以减少阻塞等待时间，但并行数目过多也会导致并发处理问题 |
| S4 | 队列分类 | 性能 | 对不同的broker分别构建请求队列，可以提高处理效率，请求队列的大小决定了耗费资源的多少，影响到broker处理性能 |

权衡点

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 决策 | 质量属性 | 分析 |
| T1 | 模块细化 | 性能，可修改性。 | 将Broker分为不同层级，可以提升系统的并发能力，但是增加了系统组件之间的通讯和系统复杂度 |
| T2 | 检测请求模式 | 安全性，性能 | 检测请求可以避免抢票插件不正常的操作数据，但是增加了额外的检测模块，降低了每次请求的响应时间 |
| T3 | 检测服务拒绝情况 | 安全性，性能 | 同上 |
| T4 | 识别信息一致性 | 安全性，性能 | 能够保证用户数据的安全与一致，但增加了需要传输的数据总量，在带宽一定的情况下降低性能 |
| T5 | 最小化系统的攻击面 | 安全性，可用性 | 请求必须经过安全处理模块，能够集中处理所有的攻击，但有可能导致单点失效 |
| T6 | Heartbeat | 可用性，性能 | 发送Heartbeat的频率决定了检测错误的时间，但是过高的频率可能消耗过多的处理资源和带宽 |
| T7 | 攻击发生时收回数据访问权限 | 安全性，可用性 | 侦测到攻击发生时，服务器模块拒绝外界访问直到确认安全，但是降低了可用性 |
| T8 | 服务定位 | 互操作性，性能 | 采用目录命名机制，方便了与外界系统的交互行为，但维护服务目录需要消耗大量的资源 |
| T9 | 模块分解 | 可修改性，可维护性 | 分解可以独立模块的职责，使得它们可以独立修改，但过多的模块数目会增加系统的层次与复杂度 |

3.风险决策

|  |  |  |
| --- | --- | --- |
| 编号 | 决策 | 风险分析 |
| R1 | 最小化系统的攻击面 | 安全处理模块能够集中处理所有的攻击，但如果安全模块收到集中攻击，可能导致单点失效 |
| R2 | Heartbeat | Heartbeat频率可以影响系统检测到错误的时间，频率过低可能导致系统无法在规定的时间内发现错误 |
| R3 | 被动冗余 | 被动冗余的broker数目过少会导致系统在错误发生时不能够及时恢复正常操作状态 |
| R4 | 攻击发生时收回数据访问权限 | 攻击发生后系统在一段时间内收回数据访问权限，导致用户和外部系统无法请求系统资源，相当于系统下线 |

团队分工

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 工作 |
| 孙康 | 141250117 | 编写安全性与可用性系统操作场景，参与Broker架构的ADD设计，具体负责服务器模块与请求总线模块的迭代设计，绘制相应模块的UML类图。搭建系统原型的主要接口与基本架构。 |
| 张文玘 | 141250192 | 编写互操作性与易用性系统操作场景，参与Broker架构的ADD设计，具体负责user-broker模块的迭代设计，绘制相应模块的UML类图。负责最终文档汇总与排版。 |
| 周小帆 | 141250209 | 编写性能系统操作场景，参与Broker架构的ADD设计，具体负责安全模块的迭代设计，绘制Broker设计的C&C视图，绘制相应模块的UML类图，负责最终文档汇总与排版。 |
| 王嘉琛 | 141250137 | 编写scalability系统操作场景，参与Broker架构的ADD设计，具体负责检测模块的迭代设计，绘制Broker设计的Module视图，绘制相应模块的UML类图。搭建系统原型的主要接口与基本架构。 |
| 吴嘉荣 | 141250148 | 参与SOA架构的ADD设计，具体负责ESB模块的迭代设计，负责绘制SOA设计的C&C视图。负责编写项目总结与经验分析部分的文档。 |
| 余旻晨 | 141250177 | 参与SOA架构的ADD设计，具体负责第一次整体迭代与购票模块的迭代设计，负责整理SOA架构设计和评估。负责编写两个体系结构优缺点分析与决策。 |
| 王梦麟 | 141250140 | 汇总系统主要功能, 参与SOA架构的ADD设计，具体负责列车时刻模块的迭代。绘制SOA设计的module视图，负责utility tree、敏感点、权衡点的编写。负责安排团队分工协作 |