**软件架构实践（第四版）\_读书笔记\_4**

第四章·可用性

*“技术并不总是既完美又可靠。 现实可能恰恰相反！”*

—Jean-Michel Jarre 让-米歇尔·雅尔

（1）可用性定义：可用性指软件在需要时能够执行任务的特性，包括可靠性和系统自我修复的能力。

（2）可用性相关概念：可用性基于可靠性，增加了恢复的概念，即系统故障时的自我修复。包括系统屏蔽或修复故障的能力，确保服务中断不超过指定时间。与鲁棒性和其他质量属性相关，涉及不可接受的失效概念。

（3）失效与故障：失效指系统与其规范的偏差，需要外部观察者确定。故障是失效的原因，可以是内部或外部的，错误是故障与失效之间的中间状态。

（4）可用性策略的三个目的：

故障检测：识别系统何时发生故障。

故障恢复：在发生故障后恢复系统服务。

故障预防：采取措施避免故障发生。

（5）故障检测策略：

①监控（Monitor）：监控系统各部分的运行状况，如处理器、进程、I/O、内存等。检测网络或其他共享资源中的失效或拥塞。

②Ping/Echo：异步请求/响应消息对在节点间交换，用于确定网络路径的可达性和延迟。需要设置时间阈值以判断组件是否失败。

③心跳（Heartbeat）：定期消息交换，检测信号，如看门狗计时器重置。

④时间戳（Timestamp）：检测分布式消息传递系统中不正确的事件序列。

⑤状态监控（Condition monitoring）：检查进程或设备中的条件，防止错误行为，如计算校验和。

⑥健全性检查（Sanity checking）：检查组件操作或输出的有效性，常用于接口。

⑦投票（Voting）：比较多个来源的计算结果，决定使用哪些结果。包括复制、功能冗余和分析冗余。

1. 故障恢复策略：

①冗余备件（Redundant spare）

主组件故障时，备用组件介入。

②回滚（Rollback）

系统恢复到先前已知的良好状态。

③异常处理（Exception handling）

处理检测到的异常。

④软件升级（Software upgrade）

服务中升级可执行代码映像。

⑤重试（Retry）

假设失效是暂时性的，重试操作可能成功。

⑥忽略错误行为（Ignore faulty behavior）

忽略特定来源的虚假消息。

⑦优雅降级（Graceful degradation）

保持关键系统功能，放弃次要功能。

⑧重新配置（Reconfiguration）

重新分配职责给正常运行的资源或组件

1. 故障预防策略：

①从服务中删除（Removal from service）

暂时停止服务以清除潜在故障。

②事务（Transactions）

确保分布式组件间交换的异步消息具有ACID属性。

③预测模型（Predictive model）

结合监控使用，预测故障并采取纠正措施。

④异常预防（Exception prevention）

技术用于防止系统异常。

⑤增加能力集（Increase competence set）

设计组件以处理更多情况，作为正常运行的一部分。

（8）可用性架构模式

冗余备件策略

①主动冗余（热备）：所有节点并行接收和处理相同 输入，冗余备件与活动节点状态同步。适用于需要快 速故障恢复的场景。

②被动冗余（温备）：只有活动成员处理输入流量， 定期向冗余备件提供状态更新。平衡了计算密集度和 成本。

③备用（冷备）：冗余备件在故障发生前保持停止服 务，需要较长的平均修复时间。不适合高可用性要求 的系统。

***冗余备件的好处与代价***

好处：系统在失效后能快速恢复正常运行。

代价：提供备件带来的额外成本和复杂性。

***其他可用性模式***

①三重模块冗余（TMR）：三个组件执行相同任务， 通过投票逻辑检测不一致。易于理解和实施，但需要 在复制级别和可用性间权衡。

②断路开关：阻止无休止的重试循环，防止系统级联 失效。需要谨慎选择超时值以避免不必要的延迟或误 报。

③进程对：使用检查点和回滚，备份在失效时接管。

④前向错误恢复：通过向前移动到理想状态摆脱不良 状态，依赖于内置的纠错功能。