系统架构设计师

DESIGNER:王川林

系统可靠性分析与设计

课程内容

- 系统故障模型 ★
- 系统可靠性分析 ★ ★ ★ ★
- 系统容错 ★ ★ ★ ★

系统故障模型

硬件的物理改变

硬件或软件中的错误状态

故障在程序或数据结构中的具体位置

失效
故障
错误

表现形式

永久性
间歇性
瞬时性

瞬时和间歇故障已经成为系统中的一个主要错误源

元件的输出线的逻辑值恒等于输入线的逻辑值

短路故障
开路故障
桥接故障

元件的输出线悬空

两条不应相连的线连接在一起

逻辑级的故障模型
逻辑结构级的故障
软件故障和软件差错
系统级的故障模型

独立差错
算术差错
单向差错

非法转移
误转移
死循环
空间溢出
数据执行
无理数据

系统级的故障模型故障在系统级上的表现为功能错误，即系统输出与系统设计说明的不一致

系统可靠性分析—可靠性指标

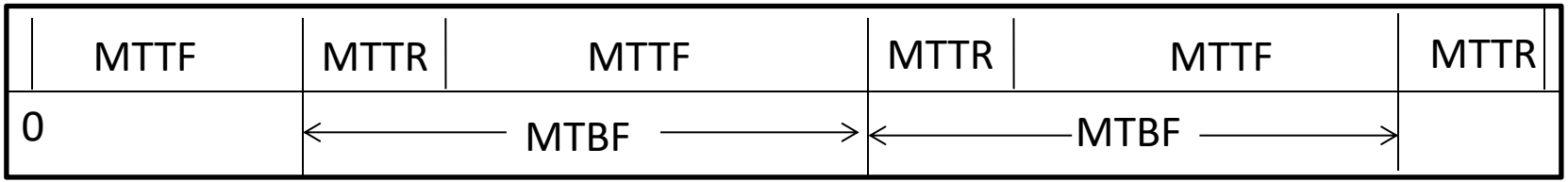
- 平均无故障时间（MTTF）
  mean time to failure
- 平均故障修复时间（MTTR）
  mean time to repair
- 平均故障间隔时间（MTBF）
  mean time between failure
- 系统可用性

MTTF=1/λ， λ为失效率

MTTR=1/μ， μ为修复率

MTBF=MTTR+MTTF

MTTF/（MTTR+MTTF）×100%



在实际应用中，一般MTTR很小，所以通常认为MTBF≈MTTF

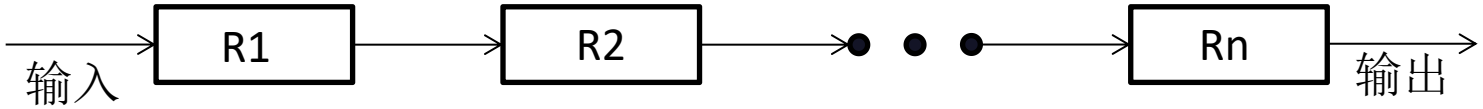
系统可靠性是系统在规定的时间内及规定的环境条件下，完成规定功能的能力，也就是系统无故障运行的概率。

系统可用性是指在某个给定时间点上系统能够按照需求执行的概率

提高可靠性需要强调减少系统中断（故障）的次数，提高可用性需要强调减少从灾难中恢复的时间　软件可靠性不等于硬件可靠性（复杂性、物理退化、唯一性、版本更新周期）

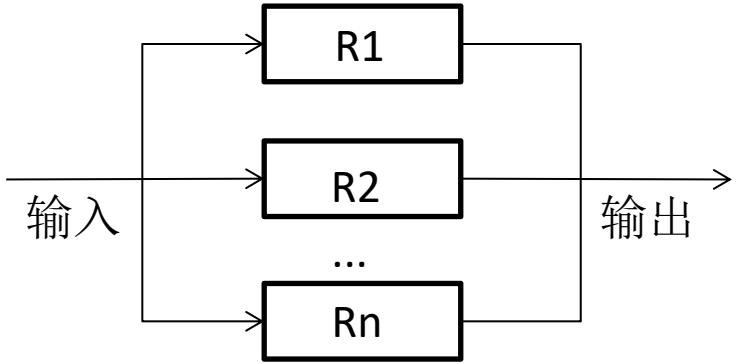
例如，假设同一型号的1000台计算机，在规定的条件下工作1000小时，其中有10台出现故障。
这种计算机千小时的可靠度R为（1000-10）/1000=0.99；
失效率为10/（1000×1000）=1×10^{-5}；
MTTF=1/（1×10^{-5}）=10^5小时。

系统可靠性分析—串联系统与并联系统



$R=R_1 \times R_2 \times \ldots \times R_n$

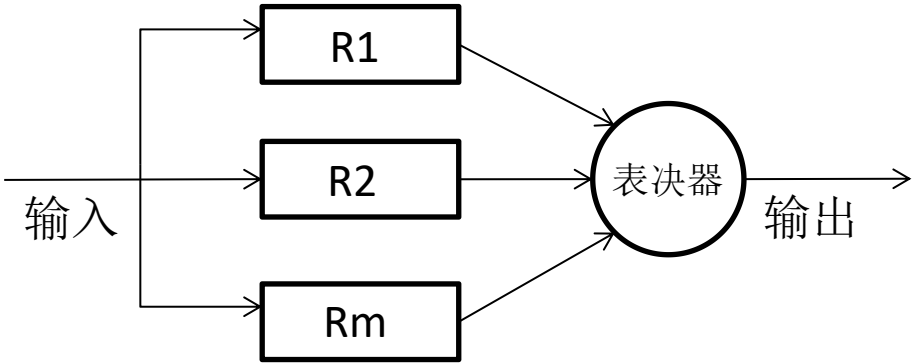
近似 $\lambda=\lambda_1+\lambda_2+\ldots+\lambda_n$



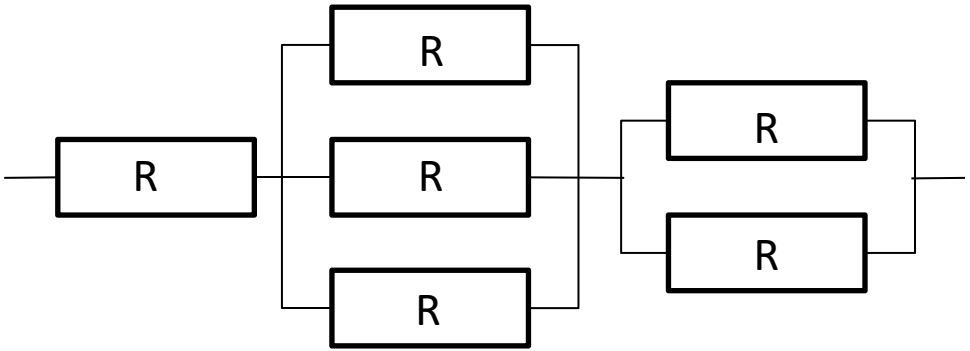
$R=1-(1-R_1)\times(1-R_2)\times\ldots\times(1-R_n)$

$$\mu=\frac{1}{\frac{1}{\lambda}\sum_{j=1}^{n}\frac{1}{j}}$$

系统可靠性分析—模冗余系统与混合系统



$$R = \sum_{i=n+1}^{m} C_m^j \times R_0^i (1 - R_0)^{m-i}$$



R×（1-（1-R）³）×（1-（1-R）²）

避错技术
查错技术
改错技术
容错技术

静态冗余(屏蔽冗余、被动冗余；模冗余系统)
动态冗余(主动冗余；备份冗余、集群系统)
混合冗余

结构冗余(硬件冗余、软件冗余)
信息冗余(校验码)
时间冗余(重复多次进行相同的计算)
冗余附加(为实现上述冗余技术所需的资源和技术）

故障检测
故障定位
故障恢复

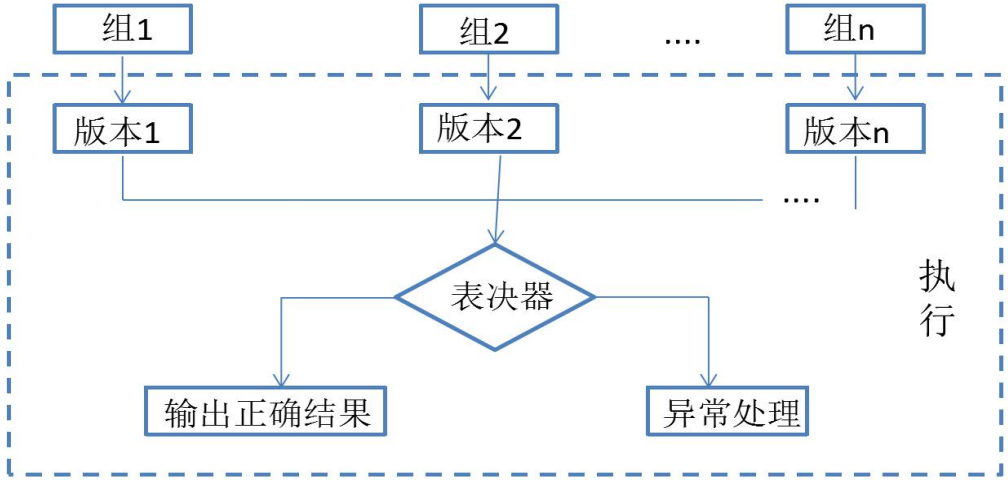
处理故障的步骤：

故障检测
故障屏蔽
故障限制
复执
故障诊断
系统重配置
系统恢复
系统重新启动
修复
系统重组合

- 前向恢复：使当前的计算继续下去，把系统恢复成连贯的正确状态，弥补当前状态的不连贯情况
- 后向恢复：系统恢复到前一个正确状态，继续执行

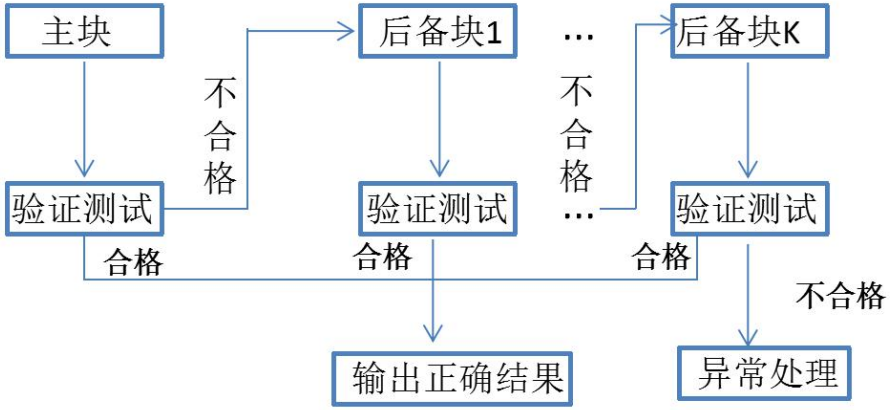
- 后向恢复简单地把变量恢复到检查点的取值；前向恢复将对一些变量的状态进行修改和处理，且这个恢复过程将由程序设计者设计
- 前向恢复适用于可预见的易定义的错误；后向恢复可屏蔽不可预见的错误

系统容错—软件容错—N版本程序设计



- ✓ 与通常软件开发过程不同的是，N版本程序设计增加了三个新的阶段：相异成份规范评审、相异性确认，背对背测试
- ✓ N版本程序的同步、N版本程序之间的通信、表决算法（全等表决、非精确表决、Cosmetie表决）、一致比较问题、数据相异性

系统容错—软件容错—恢复块方法



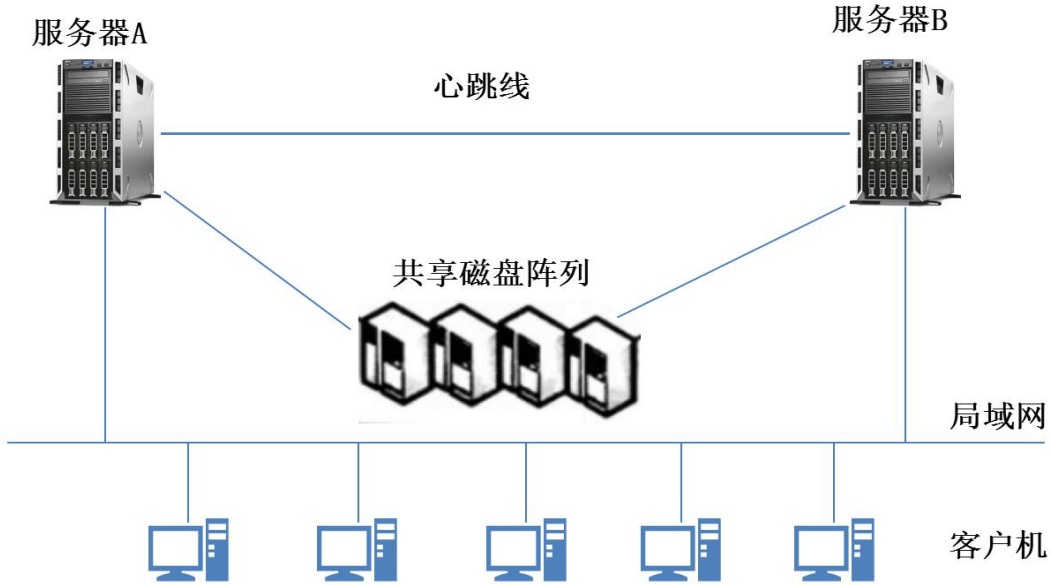
- 设计时应保证实现主块和后备块之间的独立性，避免相关错误的产生，使主块和备份块之间的共性错误降到最低程度。
- 必须保证验证测试程序的正确性。

	恢复块方法	N版本程序设计
硬件运行环境	单机	多机
错误检测方法	验证测试程序	表决
恢复策略	后向恢复	前向恢复
实时性	差	好

- 对于程序中存在的错误和不一致性，通过在程序中包含错误检查代码和错误恢复代码，使得一旦错误发生，程序能撤销错误状态，恢复到一个已知的正确状态中去

- 实现策略：错误检测、破坏估计、错误恢复

系统容错—双机容错



双机热备模式（主系统、备用系统）
双机互备模式（同时提供不同的服务，心不跳则接管）
双机双工模式（同时提供相同的服务，集群的一种）

高性能
主机系统

PK

集群系统

可伸缩性
高可用性
可管理性
高性价比
高透明性

阅读以下信息系统可靠性问题的说明，在答题纸上回答问题1至问题3.

【说明】

软件公司开发一项基于数据流的软件，其系统的主要功能是对输入的数据进行多次分析、处理和加工，生成需要的输出数据。需求方对该系统的软件可靠性要求很高，要求系统能够长时间无故障运行。该公司将该系统设计交给王工负责。王工给出该系统的模块示意图如图12-18所示。王工解释：只要各个模块的可靠度足够高，失效率足够低，则整个软件系统的可靠性是有保证的。

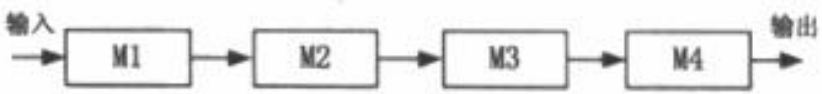


图 12-18　王工建议的软件系统模块示意图

李工对王工的方案提出了异议。李工认为王工的说法有两个问题：第一，即使每个模块的可靠度足够高，假设各个模块的可靠度均为0.99，但是整个软件系统模块之间全部采用串联，则整个软件系统的可靠度为0.99=0.96，即整个软件系统的可靠度下降明显；第二，软件系统模块全部采用串联结构，一旦某个模块失效，则意味着整个软件系统失效。

李工认为，应该在软件系统中采用冗余技术中的动态冗余或者软件容错的N版本程序设计技术，对容易失效或者非常重要的模块进行冗余设计，将模块之间的串联结构部分变为并联结构，来提高整个软件系统的可靠性。同时，李工给出了采用动态冗余技术后的软件系统模块示意图，如图12-19所示。

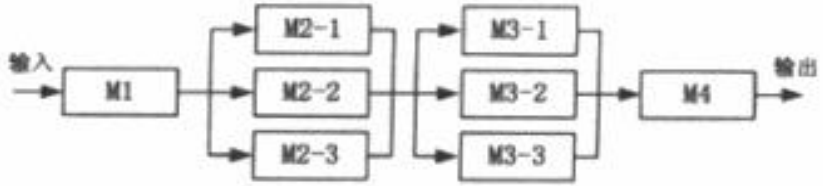


图 12-19　李工建议的软件系统模块示意图

刘工建议，李工方案中M1和M4模块没有采用容错设计，但M1和M4发生故障有可能导致严重后果。因此，可以在M1和M4模块设计上采用检错技术，在软件出现故障后能及时发现并报警，提醒维护人员进行处理。

注：假设各个模块的可靠度均为0.99

【问题1】（4分）
在系统可靠性中，可靠度和失效率是两个非常关键的指标，请分别解释其含义。

【问题2】（13分）
请解释李工提出的动态冗余和N版本程序设计技术，给出图12-18中模块M2采用图12-19动态冗余技术后的可靠度。
请给出采用李工设计方案后整个系统可靠度的计算方法，并计算结果。

【问题3】（8分）
请给出检错技术的优缺点，并说明检测技术常见的实现方式和处理方式。

【问题1】（4分）
可靠度就是系统在规定的条件下、规定的时间内不发生失效的概率。

失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

【问题2】（13分）
动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来替代它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可以不工作。前者叫冷备份系统（双工系统、双份系统），后者叫热备份系统（双重系统）。

N版本程序设计是一种静态的故障屏蔽技术，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。

M2采用动态冗余后的可靠度为：
$R=1-(1-0.99)^3 \approx 0.999999$

李工的方案同时采用了串联和并联方式，其计算方法为首先计算出中间M2和M3两个并联系统发的可靠度，再按照串联系统的计算方法计算出整个系统的可靠度。

$R=0.99*0.999999*0.999999*0.99\approx 0.98$

【问题3】（8分）
检错技术实现的代价一般低于容错技术和冗余技术，但有一个明显的缺点，就是不能自动解决故障，出现故障后如果不进行人工干预，将最终导致软件系统不能正常运行。

检错技术常见的实现方式：最直接的一种实现方式是判断返回结果，如果返回结果超出正常范围，则进行异常处理；计算运行时间也是一种常用技术，如果某个模块或函数运行时间超过预期时间，可以判断出现故障；还有置状态标志位等多种方法，自检的实现方式需要根据实际情况来选用。

检错技术的处理方式，大多数都采用“查处故障-停止软件运行-报警”的处理方式。但根据故障的不同情况，也有采用不停止或部分停止软件系统运行的情况，这一般由故障是否需要实时处理来决定。