计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(7):129−137 [doi: 10.15888/j.cnki.csa.009176]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

面向拜占庭弹性的余度管理方案①

左　力, 卿　宸

(中国电子科技集团公司第十研究所 航空电子系统重点技术实验室, 成都 610036)
通信作者: 左　力, E-mail: 2696697322@qq.com

摘　要: 综合化航空电子系统是新一代飞机的一个重要特征, 其可靠性和稳定性对整个飞机的飞行和安全起着决定性作用. 针对航电系统应当具有高可靠性的特点, 提出一种分布式集群余度架构, 并设计相应的余度管理方法, 以容忍航电系统故障后可能出现的拜占庭错误, 有效提高容错计算机的可靠性和容错能力. 采用门限签名和集群选主两种方案优化提出的余度管理方法, 降低集群中余度计算机之间的通信开销, 避免影响航电系统的实时性, 提高余度管理效率. 通过模拟实验进行测试, 结果验证了分布式集群余度管理方法可以有效提升航电系统的可靠性, 增强拜占庭弹性, 实现在 n 余度的航电系统中只要拜占庭节点数小于 $n/3$, 系统仍然能够正确运行, 并且优化方案具有更低的通信开销和计算开销.

关键词: 航电系统; 分布式结构; 拜占庭弹性; 容错计算机; 余度管理

引用格式: 左力,卿宸.面向拜占庭弹性的余度管理方案.计算机系统应用,2023,32(7):129–137. http://www.c-s-a.org.cn/1003-3254/9176.html

Byzantine Resilience Oriented Redundancy Management Scheme

ZUO Li, QING Chen

(Key Technology Laboratory of Avionics System, the 10th Research Institute of China Electronics Technology Group Corporation, Chengdu 610036, China)

Abstract: Integrated avionics system is an important feature of the new generation of aircraft, and its reliability and stability play a decisive role in the flight and safety of the entire aircraft. As the avionics system should possess high reliability, a distributed cluster redundancy architecture is proposed, and the corresponding redundancy management scheme is designed to tolerate Byzantine errors that may occur after avionics system failure and effectively improve the reliability and fault tolerance of fault-tolerant computers. The proposed redundancy management scheme is optimized by the two schemes of threshold signature and cluster selection to reduce the communication overhead between redundancy computers in the cluster, avoid affecting the real-time performance of the avionics system, and improve the redundancy management efficiency. Through simulation experiments, the results verify that the distributed cluster redundancy management scheme can effectively improve the reliability of the avionics system and enhance Byzantine resilience. Meanwhile, in an n-redundancy avionics system, the system can still operate correctly as long as the number of Byzantine nodes is less than $n/3$, and the optimization scheme has lower communication and computing costs.

Key words: avionics system; distributed structure; Byzantine resilience; fault-tolerant computer; redundancy management

自 20 世纪 90 年代以来, 综合化模块化航空电子 (integrated modular avionics, IMA) 系统一直是学术界和相关工业界的研究热点. 航电系统在实现飞机运行中信息的测量、采集、传输、处理、监控、显示等方

① 收稿时间: 2022-12-04; 修改时间: 2023-01-17; 采用时间: 2023-02-27; csa 在线出版时间: 2023-05-19
CNKI 网络首发时间: 2023-05-23

面占有重要地位, 对于保障飞行安全具有重要意义[1].

航电系统是一个安全关键系统 (safety critical system, SCS), 其失效可能导致生命财产损失或者国家安全威胁等重大或灾难性后果[2], 因此航电系统的可靠性和安全性必须要得到保障. 除此之外, 航电系统需要使用不同的任务设备完成不同任务环境下的工作, 因此航电系统还应当具有良好的可扩展性和易维护性.

针对航电系统的可靠性和安全性问题, 目前国内外普遍采用容错技术来达成目的. 容错技术通过一系列手段实现对故障的检查与屏蔽, 从而使整个航电系统的平均无故障时间 (mean time between failure, MTBF) 得到提升[3]. 余度技术是实现容错的一种重要方法, 可靠性较低的组件通过余度技术构建出高可靠性或超高可靠性的系统[4]. 因此在航空航天飞行控制、空中交通管制等重要领域具有广泛的应用.

针对航电系统的可扩展性与易维护性问题, 分布式技术是解决该问题的不二选择. 近年来, 分布式架构在航电系统中得到了越来越广泛的研究与应用, 也是采用硬件冗余的飞控计算机的主要选择[5]. 随着资源数目的增加, 分布式架构中各节点之间的相互依存关系变得越来越复杂, 任何一个子节点的故障都有可能导致整个系统的崩溃. 因此, 采用分布式架构的容错计算机必须对系统的余度管理方案进行精心设计, 才能有效提高自身的可靠性.

为了使航电系统中的容错计算机对于安全关键或者任务关键应用足够可靠, 它必须将故障考虑为包括错误构件的任意行为, 即拜占庭故障[6]. 它包括故障构件停止工作或故障构件在未来某个时刻重启, 并发送矛盾信息到不同的目的地等在失效构件企图破坏系统的能力范围内的任何情况[7].

为适应和推动航电系统的发展, 本文提出不限制失效构件任意行为的任何事先假设, 并且具有较强可扩展性的余度管理办法. 该架构比采用传统的失效模式与影响分析的架构解决发生没有被覆盖的失效模型问题所耗费的成本和时间要少得多, 并且实现拜占庭弹性要求的运行时开销要大大少于使用基于限制失效行为模型的容错技术达到故障覆盖水平的开销.

本文的主要研究内容如下.

(1) 构建分布式集群余度结构, 并提出相应的余度管理方法, 不采用主流的主从备份结构和多数表决结构, 实现对拜占庭弹性的需求, 提高航电系统的可靠性和安全性.

(2) 采用集群选主和门限签名两个方案, 优化分布式集群余度管理方法, 在一定程度上减少集群中余度计算机之间的通信消耗, 加快容错计算机达成一致的效率.

本文第 1 节对航电系统中的容错计算机结构和门限签名算法进行介绍; 第 2 节提出分布式集群余度结构和相应的余度管理方法, 以达到增强拜占庭弹性的目的; 第 3 节针对提出的余度管理方法在通信开销和计算开销上的缺陷, 采用集群选主方案和门限签名方案进行优化; 第 4 节通过实验模拟本文提出的余度管理方法, 证明该方案的有效性并验证优化后方法的有效改进; 最后, 对本文主要工作和贡献进行简要总结.

1 相关基础

1.1 容错计算机结构

余度管理是余度技术的重要环节, 主要用于管理系统中冗余资源的选择、配置和工作方式, 其目的就是在最大程度上提高冗余资源的利用率, 加强系统进行动态重构时策略选择上的灵活性, 最终使系统具有最大的容错能力[8]. 如果航电系统的余度管理方案设计不当, 反而可能导致可靠性变得更差.

容错计算机余度管理结构一般为主从备份结构、多数表决结构或两种结构的结合. 主从备份结构[9] 中, 若干能够实现相同功能的余度计算机都具有驱动执行机构的能力, 但只有一个按照固定优先级选出来的优先级最高的主计算机允许输出, 控制执行机构动作, 其余计算机为备份计算机, 如图 1 所示.

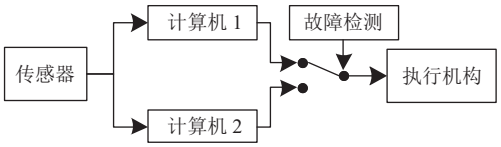

图 1 主从备份结构

主从备份结构实现容错最关键的要素是故障检测技术, 它能够及时且正确地检测出故障, 并切换到备份计算机. 常用的故障检测技术包含机内自检测技术和自检测对比监控技术, 但绝大多数都难以达到 100% 的自检覆盖率.

多模余度表决结构[10] 运用的是故障掩盖技术, 通

过表决算法对至少 3 个通道中并列运行的余度计算机的输出结果进行表决, 可以采用多数表决或中值选择等表决方式. 多数表决方式对所有通道输出进行比较, 取多数者作为正确结果[11], 如图 2 所示.

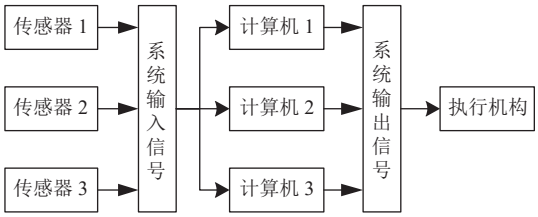

图 2 多模余度结构

但多模余度系统中可能存在潜在故障, 并且不会检测和排查系统故障, 所以需要结合附加的方法进行故障检测和隔离.

本文提出分布式集群余度结构, 该结构类似于多模余度表决结构, 但其不需要使用硬件或者软件的表决器对输入输出信号进行表决, 而是依赖余度计算机间的通信赋予系统抵抗拜占庭错误的能力.

1.2 门限签名

门限签名是结合门限秘密共享技术的一种特殊的数字签名. 简单来说就是使用 Shamir 秘密分享技术将一对公私钥 (P, K) 分为多个不同的私钥分量和可验证密钥集. (t, n) 门限签名方案表示在有 n 个签名者的集群中, 必须要由至少 t 个合法成员才能代表群体签名, 并用公钥 P 进行验签.

BLS 算法[12] 是一种常用的门限签名算法, 它是基于双线性映射构造的一种加密算法, 主要包含 4 部分: 算法初始化、密钥生成、数据签名和数据验签.

算法初始化: 设 G_1 和 G_2 是阶为 p 的乘法循环群, 其生成元分别是 g_1 和 g_2, 双线性映射 e 表示为 $G_1\times G_2\to G_T$, H: $\{0,1\}^*\to G_1$ 表示一个安全哈希函数, $(G_1, G_2, G_T, e, g_1, g_2, p, h)$ 是公开参数.

密钥生成: 设定签名门限为 t, 选择一个随机数 s $(s\in Z_p)$, 计算 $q=g_2^s\in G_2$, 得到私钥 $SK=x$, 公钥 $PK=q$. 随机选一个 Z_p 上的 $t-1$ 阶多项式 P, 满足 $P(0)=s$, 计算节点 i 的私钥 $s_i=P(i)$, 公钥 $q_i=g_2^{s_i}\in G_2$, 向集群广播公钥 q 和密钥集 $\{q_1, q_2, \cdots, q_n\}$.

签名过程: 节点 i 对消息 m 的部分签名为 $sig_i=H(m)^{s_i}$, 多个部分签名计算完整门限签名的方法如公式:

$$shareSig=\prod_{i=1}^{t} sig_i^{\lambda_i} \tag{1}$$

其中,

$$\lambda_i=\frac{\prod\limits_{j=1,\ j\neq i}^{t}(0-j)}{\prod\limits_{i=1,\ j\neq i}^{t}(i-j)} \bmod p \tag{2}$$

验证过程: 对部分签名 sig_i 和消息 m 验证过程为判断 $e(g_2, sig_i)=e(q_i, H(m))$ 是否成立, 对门限签名 $shareSig$ 的验证过程为判断 $e(g_2, shareSig)=e(q, H(M))$ 是否成立.

门限签名技术适用于签名次数较少, 验签次数很多的场景, 本文提出的余度管理办法中使用门限签名算法可以有效减少通信开销和计算开销.

2 方案设计

2.1 总体架构

本文提出采用分布式集群余度结构提高航空电子系统的可靠性和扩展性, 如图 3 所示. 由 4 个余度计算机构成一个分布式集群, 节点间互相配合输出正确且一致的命令控制执行机构行动. 当发生拜占庭错误时, 集群可以通过相应的余度管理办法实现容错.

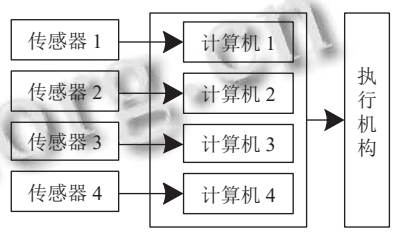

图 3 分布式集群余度结构

针对计算机与传感器的连接方式主要有两种.

(1) 每个传感器都与各余度计算机连接, 该连接模型性能最好, 但对余度计算机的要求以及资源消耗也很高, 由于目前绝大多数航电系统都采用的是总线结构, 即所有传感器和计算通道都在一条总线上, 因此航电系统中更有很可能采用该种连接方式.

(2) 各传感器与各计算机一一对应, 一个余度计算机只采集一组传感器数据, 余度计算机之间进行数据交换和表决, 该模型不消耗额外资源, 但需要通过余度管理办法避免与传感器连接的通道出现故障的情况.

采用分布式集群架构不同于主从备份结构和多数

表决结构. 主从备份结构中余度计算机之间存在固定优先级关系, 而采用分布式架构的航电系统将每个余度计算机作为一个对等节点, 共同构成一个去中心化的对等网络, 系统运行时所有计算机相互配合, 对执行机构进行一致的可靠控制.

多数表决结构中每个余度计算机都在运行, 不会造成资源的浪费, 但最终容错计算机的输出仍然需要使用表决器判断. 而分布式架构不需要使用表决器, 它通过余度计算机之间的通信达成一致的正确结果, 并且分布式架构的余度管理方案使得航电系统可以容忍拜占庭错误.

2.2 余度管理方法

本文提出采用分布式集群架构实现高可靠的航电系统, 但如果余度管理方案设计不当也无法使航电系统具有高可靠性. 为说明本文提出的余度管理办法具有一般性, 以只有一个传感器连接到通道 A 的四余度结构为例说明分布式集群余度结构相应的余度管理办法, 如图 4 所示.

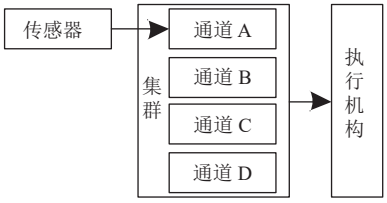

图 4 单传感器连接方式

分布式集群余度管理办法的过程可分为 request 阶段、pre-prepare 阶段、prepare 阶段、commit 阶段和 reply 阶段, 共 5 个阶段, 如图 5 所示.

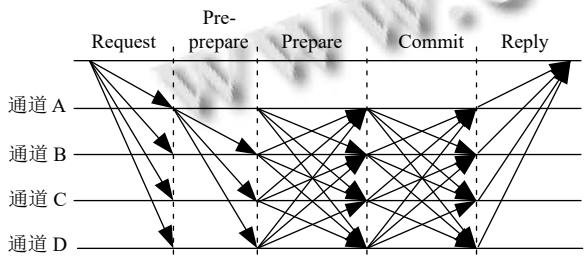

图 5 余度管理办法

(1) Request 阶段: 传感器向分布式集群中与自己连接的通道 A 发送数据.

(2) Pre-prepare 阶段: 通道 A 接收到来自传感器的数据后, 向集群广播消息.

(3) Prepare 阶段: 各通道的余度计算机对收到的数据进行校验, 判断数据是否在正常范围中, 如果检测没有问题, 则广播 prepare 消息. 等待收集到 prepare 消息数量达到 $2f$+1 门限时, 计算机将会向集群中广播 commit 消息.

(4) Commit 阶段: 各余度计算机向其他通道发送 commit 消息的同时, 也接受来自其他通道的 commit 消息. 若收集到经过了校验的 $2f$+1 个 commit 消息, 则表示集群内部已经达成一致的输出.

(5) Reply 阶段: 当输出端接收到 f+1 个 reply 应答就执行命令.

除此之外, 当各通道都可以获取全部连接器的数据时, 也可以极大地简化余度管理过程, 如大中型航电系统采用总线架构可以使所有通道单独获取传感器传送到总线上的数据, 如图 6 所示. 该连接方式下 prepare 阶段的作用由传感器与通道之间的复杂连接完成, 因此通过 3 阶段就可以达成一致, 避免拜占庭故障的影响.

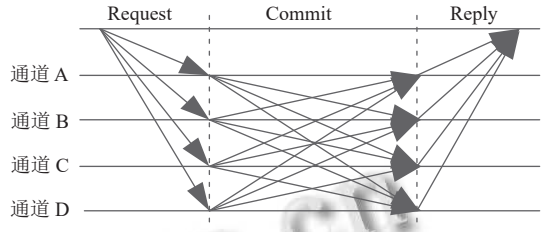

图 6 多连接方式下余度管理办法

2.3 安全性分析

根据分布式余度集群与传感器的连接方式不同, 分别分析两种情况下航电系统的拜占庭弹性.

(1) 在传感器与集群中所有通道都存在连接的模型下, 每个通道直接从传感器获取数据, 整个分布式集群中所有节点都是对等节点, 此时的网络是一个去中心化的网络. 假设该模型下某节点发生拜占庭错误, 如图 7 所示为其余度管理办法过程.

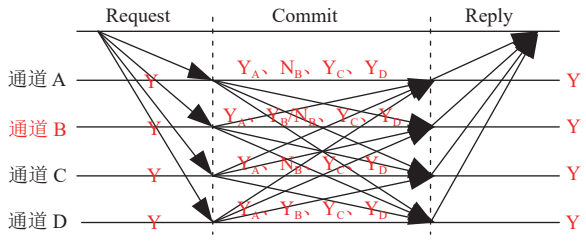

图 7 对等节点发生拜占庭故障

由图 7 可知, 节点 B 发生拜占庭错误, 向节点 A、C 发送错误消息, 向节点 D 发生正确消息, 但由于节点 A、C、D 都是正确节点, 因此节点 B 无法破坏系统安全性, 详细分析如表 1 所示.

表 1 对等节点发生拜占庭故障情况分析

阶段	发送	接收			
		A	B	C	D
Request	传感器	Y	Y	Y	Y
Commit	A	Y_A	Y_A	Y_A	Y_A
	B	N_B	Y_B	N_B	Y_B
	C	Y_C	Y_C	Y_C	Y_C
	D	Y_D	Y_D	Y_D	Y_D
结果统计		Y	Y	Y	Y

(2) 在传感器只与集群中某个通道相互连接的模型下, 本文提出的分布式余度集群管理办法需要有一个通道接收传感器数据并将其发送到分布式集群中各个通道内, 完成该功能的节点通常称为主节点, 其他通道称为从节点. 若从节点发生拜占庭故障, 其产生的影响与上述传感器与所有通道连接的情况那个相同, 因此, 在该模型下只需要讨论主节点发生拜占庭故障的情况.

在传感器单连接模型下, 主节点只能由与该传感器连接的通道担任, 但由于该主节点只起到向集群输入数据的作用, 其他时刻集群是一个完全去中心化的分布式集群, 因此不会破坏系统安全性, 即使主节点是拜占庭节点, 通过 prepare 和 commit 两个阶段的信息交互, 仍然不能对系统产生最终的正确结果产生影响.

Commit 阶段与 prepare 阶段过程相同, 但发送的消息不同, 主要解决集群中主节点发生拜占庭错误的情况. 如图 8 展示的是当主节点 A 发生拜占庭故障且余度管理办法只采用 4 阶段时的情况.

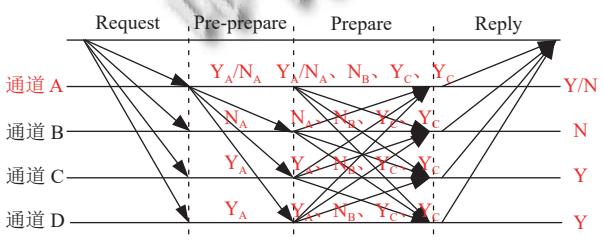

图 8 主节点发生拜占庭故障

由图 8 可知, 主节点 A 可以向不同的从节点发送不同的消息, 在 pre-prepare 阶段节点 A 向节点 B 发送错误消息 N, 因此 prepare 阶段节点 B 将向网络中广播错误消息 N, 同理, 节点 C、D 在 prepare 阶段将会向网络中广播正确信息. 若 4 阶段的余度管理办法到此结束, 则节点 B 将会得到错误结果, 节点 C、D 将会得到正确结果, 则拜占庭节点 A 破坏了系统的完整性, 正确节点得到了不一样的结果, 详细分析如表 2 所示.

表 2 主节点拜占庭故障情况分析

阶段	发送	接收			
		A	B	C	D
Pre-prepare	A	—	N_A	Y_A	Y_A
Prepare	A	—	N_A	Y_A	Y_A
	B	N_B	N_B	N_B	N_B
	C	Y_C	Y_C	Y_C	Y_C
	D	Y_D	Y_D	Y_D	Y_D
结果统计		Y/N	N	Y	Y

基于此分析, 当余度管理办法采用 4 阶段时主节点出现拜占庭故障, 航电系统将不再具有安全性和一致性, 因此需要 5 阶段的余度管理办法, 在 commit 阶段正确节点之间将会相互交换信息, 这时正确节点将发现主节点发送的消息不一样, 因此可以判断主节点发生了拜占庭故障.

只有在确保与传感器连接的余度计算机确定是可靠的情况下, 余度管理办法的 5 阶段才可以合并为 4 阶段, 如图 9 所示为简化为 4 阶段的余度管理办法.

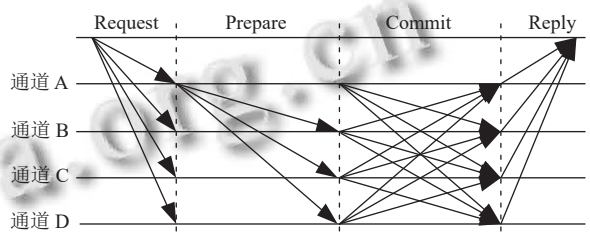

图 9 简化余度管理办法

去掉 pre-prepare 阶段是由于 pre-prepare 阶段和 prepare 阶段的主要有两个作用: 一是避免主通道发生拜占庭错误, 向其他通道发送不一样的信息; 二是完成所有通道都能够获得传感器数据, 需要主通道将自己获得的数据发送给其他通道. 而在确定主通道不会发生拜占庭错误的情况下, pre-prepare 阶段和 prepare 阶段就只存在发送信息给所有通道的作用, 因此能够将两阶段合并为 prepare 阶段, 实现简化余度管理办法.

当主节点发生拜占庭错误时, 若没有采用选主方案, 则由于整个分布式余度集群无法获得正确的传感器数据, 将导致整个分布式余度集群无法工作. 但针对

航电系统的应用环境, 绝大多数情况下不会使用传感器单连接模型, 更多的是传感器与集群中多个余度计算机相连接. 这时当正确节点发现主节点发生拜占庭故障, 则会触发主节点选举机制, 具体过程将在第 3.1 节进行分析.

3 优化方案

分布式集群余度管理方法在提高航电系统容错能力、加强拜占庭弹性的同时, 也会增加余度计算机的计算量与通信量. 简单来说就是以计算开销和通信开销换取航电系统的可靠性与安全性. 但针对小型机载环境或要求长时间工作的环境而言, 机载资源的消耗应当尽可能节省. 因此, 本节将讨论如何在确保整个航电系统安全可靠的前提下, 降低机载资源的消耗, 减少余度计算的计算开销和通信开销.

3.1 集群选主方案

分布式集群余度管理办法通过去中心化的网络增强拜占庭弹性, 保障容错计算机即使在复杂的任务环境下发生任何破坏系统安全的行为, 航电系统仍然能够可靠稳定的运行. 而实现该功能主要依赖于去中心化网络中点对点的传输方式, 这使得航电系统的通信损耗急剧增加.

为解决通信开销问题, 采用集群选主方案进行优化. 在航电系统中选择一个余度计算机作为主节点, 由该节点保持与其他节点的通信, 减少节点间的相互通信, 并收集所有节点的表决结果, 如图 10 所示.

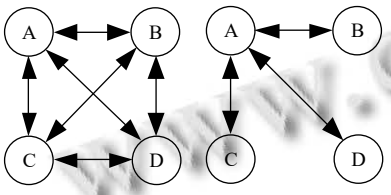

图 10 普通方案与选主方案对比

原方案中需要节点间相互发送和收集数据, 其通信量随着节点规模的增大而呈现指数级的增长, 严重影响了航电系统的实时性. 优化方案将主节点作为中心节点, 在一定程度上削弱了去中心特性, 但对于不强调去中心化的航电系统来说, 隐形中心化是能够被接受的. 优化后的余度管理方法通信量随着节点规模变化呈线性变化, 这对加快航电系统的效率具有很大作用.

如图 11 所示为采用集群选主方案后的余度管理过程. 将通道 A 作为主通道, 其余通道检验输入数据后将结果和签名发送到通道 A 中, 并将其打包为一个签名集合后再次发送到其他通道.

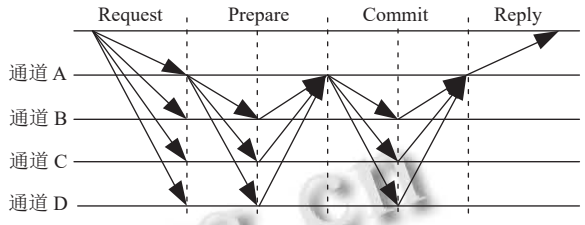

图 11 集群选主方案下余度管理方法

该过程避免了原方案中每个通道自己收集签名和结果的情况, 而改为由主通道完成, 减少通道间相互通信的开销. 但采用集群选主方案也带来了新的问题, 即主节点如何选择.

不同于原方案是一个完全去中心化的网络, 即使存在主节点, 其作用也只是接收和发送传感器数据, 而采用集群选主方案后, 隐形去中心化的网络中主节点还需要负责收集和发送签名信息, 因此需要考虑主节点的选择以及主节点发生故障时的处理方案.

当主节点发生故障或集群中的从节点认为主节点是拜占庭节点时, 就会触发主节点选举事件, 如图 12 所示为分布式余度集群下主节点选举流程, 可以分为 3 个阶段.

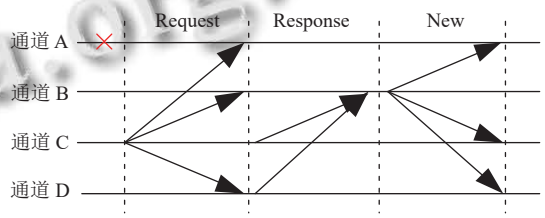

图 12 主节点选举

(1) Request 阶段: 通道 C 认为主节点故障或主节点发生拜占庭行为导致整个系统长时间不能达成一致, 则会向所有其他节点发送切换主节点请求;

(2) Response 阶段: 所有通道接收到请求切换主节点时, 将会推选当前除问题节点外存活的编号最小的节点成为新的主节点, 向其发送自己的签名, 若通道认为主节点没有问题不会响应;

(3) New 阶段: 当主节点候选节点收集到 $2f+1$ 个签名, 则证明集群中有足够多的节点认为主节点是有

问题的, 因此向大家广播签名集合作为自己成为新主节点的上任通知, 告诉所有节点主节点发生了改变.

由于存在签名门限 $2f+1$, 所以集群中大多数节点总是一致的, 具有相同的判断结果, 它们会共同推选出相同的新主节点, 这确保了系统中总是有且只有一个主节点, 保障系统安全性.

3.2 门限签名方案

门限签名虽然不能够减少节点间的通信次数, 但可以减低每次通信时的开销. 门限签名的主要作用是可以将多个签名数据压缩合并成一个门限签名. 验证者只需要对单个门限签名进行验证, 其验证结果等效于验证者对所有签名一一验证的结果. 门限签名可以节约存储空间, 减少签名者和验证者之间的通信成本, 特别适用于多个计算机之间点对点的验证场景.

对于分布式集群余度管理方案, 在进行表决时, 主通道需要构建有效数字签名集合证明前一阶段收集到了足够的签名, 这造成了较大的通信开销, 尤其是在高余度的情况下. 而且其他通道需要对集合中的签名进行一一校验, 这也会带来巨大的计算开销, 降低系统效率, 如图 13 所示.

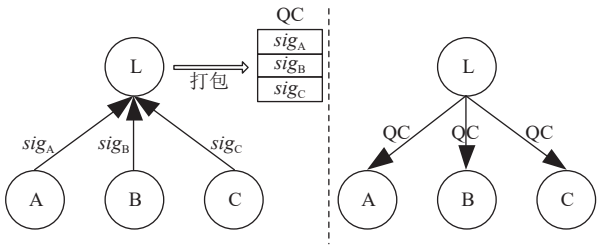

图 13 普通签名方案

采用门限签名方案代替普通签名方案, 使得主通道不需要再去构建有效签名集合, 而是通过合成门限签名作为收集到足够签名集合的依据, 保障系统的可靠性.

在 prepare 阶段和 commit 阶段收集签名时, 主通道将收集并验证其他通道的部分签名 sig_i, 在数量达到门限值 $2f+1$ 时计算完整门限签名 *shareSig*, 其他余度计算机不需要对签名集合依次验证, 而是直接使用主公钥 q 对签名 *shareSig* 的有效性进行验证.

使用门限签名方案, 能够在保证可靠性的同时, 将 prepare 消息和 commit 消息包含的至少 $2f+1$ 个签名集合减少到 1 个, 将原来至少 $2f+1$ 次验签操作降低为 1 次, 以达到降低通信量和计算量的目标, 如图 14.

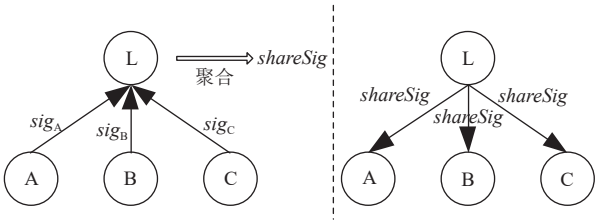

图 14 门限签名方案

目前采用门限签名技术降低通信开销或计算开销的应用非常广泛, 比如在区块链系统中常采用 ECDSA、Schnorr、BLS 等门限签名方案, 降低通信复杂度, 再比如分级 PKI 中的证书链、安全邮件、电子现金交易、数据库外包、无线传感器网络安全路由协议、车联网信息聚合、日志审计等多种现实生活中的应用, 尤其是在分布式系统和云计算当中具有广泛应用, 因此将其应用到本文采用的余度管理方法上具有很强的可行性和可实现性.

4 仿真与实验

本文以 4 余度容错计算机为例对航电系统抗拜占庭能力进行模拟, 验证提出的分布式集群余度结构是否有效, 验证相应的余度管理办法是否容忍拜占庭故障, 并获得分布式集群余度管理方法的通信开销和计算开销, 实验环境如表 3 所示.

表 3 实验环境

名称	版本/型号
操作系统	Ubuntu 16.04
容器版本	Docker 19.03.1
CPU	Intel(R) Core(TM) i5-10400 CPU @ 2.90 GHz
编程环境	go1.16.5 Linux/amd64
绘图工具	Matlab 2018a

4.1 通信开销

测试采用分布式集群余度结构的航电系统中余度管理过程所需要的通信开销. 分布式集群余度管理办法中涉及的通信开销主要是集群中余度计算机之间的表决开销. 测试结果如图 15 所示.

由实验结果可知, 针对传统的主从备份结构和多模余度结构, 其通信开销为 0, 因为在传统结构中余度计算机之间不需要通信, 其输出表决依赖于其他硬件的配合, 如表决器或故障检测, 因此无法避免拜占庭错误. 分布式集群余度结构依赖余度计算机之间的通信和校验实现拜占庭弹性, 提高系统可靠性, 因此其通信

量较高, 并且随着余度提升通信量呈指数级增长. 最后, 通过集群选主方案在一定程度上降低通信次数, 使通信开销不再呈现指数级增长, 而是随着余度提升呈线性增长.

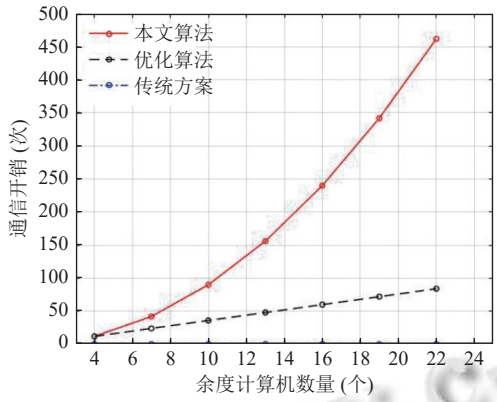

图 15 通信开销对比

4.2 计算开销

分布式集群余度管理方法除了与传统方案一样需要进行航电系统本身功能的控制律计算外, 还需要额外进行余度计算机间的签名和验签操作, 这是为了对抗拜占庭错误所必须的操作. 拜占庭行为定义为任何破坏航电系统安全性的行为, 在设计余度管理方法时也应当考虑伪造签名破坏系统的可能, 因此签名和验签操作不可避免. 计算开销测试结果如图 16 所示.

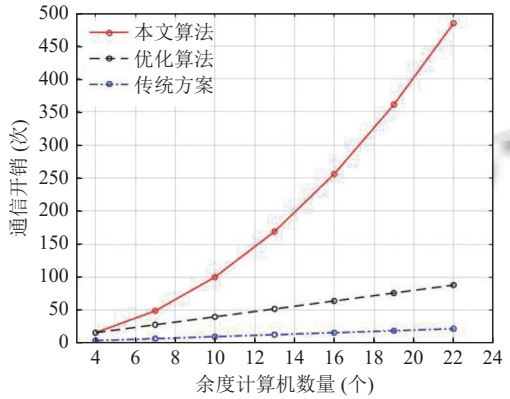

图 16 计算开销对比

观察测试结果, 传统余度控制方法中计算开销主要是每个余度控制计算机的控制律计算, 不需要额外的运算, 因此计算量与余度规模呈正比. 分布式集群余度管理方法需要余度计算机间相互的签名和验签, 其计算次数随余度增加呈指数级增长. 优化方案应用了集群选主和门限签名方式, 使余度计算机间的通信次数和验签次数大大减少, 因此其计算开销也急剧减少, 并且随余度的增加呈线性增长.

4.3 拜占庭弹性

分布式集群余度管理办法能够保障航电系统在发生拜占庭故障时, 仍然可以正常工作, 做出正确判断. 模拟航电系统中某通道发生拜占庭故障, 让该通道尝试向其他通道发送不同消息, 或者让通道不发送任何消息, 测试航电系统是否还可以正常运行. 经过多次测试, 得到如图 17 所示的测试结果.

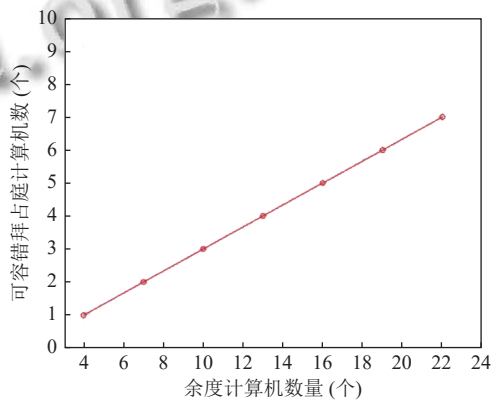

图 17 拜占庭弹性测试

为使分布式集群能够容忍拜占庭错误, 假设当前系统的余度数为 n, 其中发生拜占庭错误的通道有 f 个, 分布式集群余度管理方法的门限为 t.

则为满足系统的活性, 考虑发生存在 f 个拜占庭节点时系统仍然可用, 应当满足:

$$t < n - f \tag{3}$$

为满足系统的一致性, 即航电系统能够做出正确表决, 应当满足:

$$t + t - n \geqslant f + 1 \tag{4}$$

由此可得, 航电系统中总余度和拜占庭错误应当满足:

$$n \geqslant 3f + 1 \tag{5}$$

该证明不失一般性, 分布式余度集群中无论是主节点还是从节点发生拜占庭错误, 由于密码学安全保证了数字签名不可伪造, 也不会发生 Hash 碰撞的情况, 所以主节点收集到的签名集合确实是整个集群中每个节点本身的签名结果. 因此经过证明可知在 $3f$+1 个节点的集群中, 最多能够容忍 f 个节点发生拜占庭故障,

也证明了分布式余度集群达成一致需要收集的签名门限个数为 2*f*+1 个.

由模拟实验的结果和理论分析结论对应可知, 采用本文提出的分布式集群余度结构和相应的余度管理办法, 能够容忍航电系统中同时存在 1/3 余度的拜占庭错误通道, 即当系统中存在 *f* 个拜占庭节点时, 为使航电系统能够正常工作, 并且得到正确结果, 分布式余度集群中必须要保证至少有 3*f*+1 个节点.

5 结束语

本文针对航空电子系统应当具有高可靠性的特点, 采用分布式集群结构, 并设计相应的余度管理办法, 增加航电系统的拜占庭弹性, 提高系统的可靠性. 采用集群选主方案和门限签名方案优化余度管理办法, 降低通信开销和计算开销. 最后本文通过模拟实验验证了改进方案的有效性, 并证明了在由 *n* 个节点构成的分布式余度集群中至多允许小于 *n*/3 个节点发生拜占庭故障. 本文采用分布式架构作为解决余度管理对可靠性和扩展性需求的研究方向, 为其他拜占庭容错方案提供了更多的设计思想和设计依据.

参考文献

1 Guo QL, Liu JJ, Zhen C, *et al*. Exploration and analysis of distributed avionics. Aeronautical Computing Technique, 2014, 44(5): 121–124.

2 Liang ZZ, Zuo QQ, Zheng YQ, *et al*. Design and implementation of fourfold redundancy fault-tolerant management algorithm. Electronic Technology, 2013, 42(12): 7–9, 4.

3 Wang Y, Jia ZQ. Redundancy management designing based on high reliability fault-tolerant control system. Aeronautical Computing Technique, 2019, 49(2): 125–129.

4 Sun XX, Li WW, Jin YZ, *et al*. Application of redundant and fault-tolerant technique in flight-control system. Journal of Hebei Institute of Architectural Engineering, 2002, 20(4): 87–89.

5 Xu M, Chen X, Yuan BB. Redundancy management and bus protocol design for distributed flight control computer. Electronics Optics & Control, 2017, 24(9): 77–82.

6 Tang N, Liu ZY, Gao YN, *et al*. Design and analysis of DFT of spacecraft control system based on Byzantine fault tolerant architecture. Aerospace Control and Application, 2017, 43(5): 55–60.

7 Xiao AB, Hu MM, Ren XC, *et al*. Reliability analysis of the computer with quad-modular redundancy Byzantine fault tolerant. Aerospace Control and Application, 2014, 40(3): 41–46.

8 Ma C, Dai XD, Guo Y. Design of quadruple redundancy flight control computer and reliability analysis based on Markov model. Information & Communications, 2019, (10): 10–11.

9 Han W, Zang HW, Xie KJ. Quad-redundant fault-tolerant computer architecture and reliability analysis. Computer Engineering & Science, 2003, 25(1): 98–100.

10 Su MJ, Yan M. Research on application of redundancy computer in UAV system. Science and Technology & Innovation, 2021, (2): 179–181.

11 Zhang R, Chen X. Fault diagnosis of UAV's distributed flight control computers. Electronics Optics & Control, 2018, 25(5): 115–119.

12 Wang TT, Hou SH. Research on threshold signature scheme and its security analysis. Computer Engineering and Applications, 2018, 54(13): 123–130.

(校对责编: 孙君艳)