现实生活充满不确定性，共识也应如此！

Reginald Frank*
reginaldfrank77@berkeley.edu 加州大学伯克利分校 伯克利，加利福尼亚州，美国

Soujanya Ponnapalli*
soujanya@berkeley.edu 加州大学伯克利分校 伯克利，加利福尼亚州，美国

Octavio Lomeli
octavio.lomeli@berkeley.edu 加州大学伯克利分校 伯克利，加利福尼亚州，美国

Neil Giridharan
giridhn@berkeley.edu 加州大学伯克利分校 美国加利福尼亚州伯克利

MarcosK. A guilera
mkaguilera@gmail.com 博通 美国加利福尼亚州帕洛阿尔托

Natacha Crooks
ncrooks@berkeley.edu 加州大学伯克利分校 美国加利福尼亚州伯克利

arXiv:2602.11362v1 [cs.DC] 11 Feb 2026

摘要

现代分布式系统依赖共识协议来构建容错核心，并以此为基础构建应用程序。共识协议在特定故障模型下是正确的，即最多允许 f 台机器发生故障。我们认为，这种 f 阈值故障模型过度简化了现实世界，也限制了针对成本或性能进行优化的潜在机会。相反，我们主张采用概率性故障模型，以捕捉实践中观察到的故障的复杂性与细微差异。概率共识协议能够明确利用单台机器的故障曲线，并探索绕开传统瓶颈（例如多数派法定人数交集）的方法，从而打造更可靠、高效、经济且可持续的系统。

CCS概念

• **计算机系统组织** → **可靠性**；**可用性**。

关键词

共识，分布式系统

ACM参考格式：

Reginald Frank*, Soujanya Ponnapalli*, Octavio Lomeli、Neil Giridharan、Marcos K. Aguilera 和 Natacha Crooks。2025年。《现实生活充满不确定性，共识也应如此！》。载于操作系统热点专题研讨会（*HOTOS 25*），*2025*年*5*月*14*日至*16*日，加拿大艾伯塔省班夫。ACM，美国纽约州纽约，8页。https://doi.org/10.1145/3713082.3730374

允许免费制作本作品全部或部分的数字或纸质副本，供个人或课堂使用，前提是副本的制作或分发并非出于营利或商业利益，且副本须在首页载有本声明及完整引文。必须尊重本作品中第三方组件的版权。对于所有其他用途，请联系所有者/作者。*HOTOS 25*，班夫，艾伯塔省，加拿大
©2025 版权由所有者/作者持有。ACM ISBN 979-8-4007-1475-7/25/05https://doi.org/10.1145/3713082.3730374

1 引言

现代分布式系统依赖于一个容错核心，它提供单个无故障组件的抽象，应用逻辑则构建于其上。在大规模环境中，故障是常态，而非例外。大多数云原生数据库 [23, 72]，配置管理器 [2, 42]，去中心化平台 [6, 40, 55, 67]，以及最新的 AI 模型训练和服务平台 [68] 都以采用共识协议的容错核心 [13, 36, 44, 50, 61] 作为关键构建模块。共识协议可在特定故障模型下保证数据可靠性。故障模型反映了人们对现实世界的认知，定义了协议所依据的具体假设。对于共识，开发人员必须对网络、故障类型和故障数量作出假设。故障模型的选择至关重要，因为它会显著影响协议设计和性能。这就像汽车保险：过于乐观的故障模型会降低达成共识的成本，却可能无法反映现实，导致系统失效。举例来说，极端情况下，可以假设故障模型中不可能发生任何故障！相反，过于悲观的故障模型则可能使协议变得不必要地复杂。

正确刻画故障的现实是一项复杂的任务。因此，当今大多数系统会简化现实，并在 f-阈值 模型下保证正确性。在不超过 f 个节点故障的情况下，系统保持安全（所有节点对已提交数据达成一致）且活跃（所有新操作最终都会被提交）。节点既可能因崩溃而失效（需要崩溃容错，即 CFT），也可能因恶意入侵而失效，此时节点可以任意偏离协议（需要拜占庭容错，即 BFT）。如果发生故障的节点超过 f 个，系统不提供任何保证。重新配置可以逐步用正确节点替换故障节点，以确保永不超出故障阈值。

这种抽象简洁明了，却过于简单化：它掩盖了现代系统故障的重要方面，也难以反映当今人们对可靠性的思考方式。

*这些作者共同领导了该项目，并做出了同等贡献

故障很复杂。 f阈值模型将服务器划分为正常或故障两类，并一视同仁地对待故障：每次故障都会为当前故障计数增加一，而不会考虑到有些服务器比其他服务器更容易发生故障。实际上，所有服务器最终都会发生故障，而且每台服务器都有其独特的故障概率，具体取决于其类型 [48]，甚至取决于它在数据中心中的位置 [66]。这一概率会随着时间推移而变化，受设备使用年限 [25, 48]，、更新部署 [24]，或工作负载变化 [77]等因素影响。此外，服务器故障既非同质也非相互独立，因为故障往往会集中出现在软件部署、意外的工作负载变化或发现新漏洞之时 [5, 19, 29, 53]。

保证仅具有概率性。当前的共识协议给出的是非此即彼的保证：它们声称在故障数少于 f 时完全具备安全性和活性，但在其他情况下不提供任何保证。这并不现实。由于故障具有概率性，因此故障数始终可能超过 f。因此，没有任何共识协议能够提供强于概率安全性或概率活性的保证。存储系统已经认识到这一现实。例如，S3 将其保证表述为可用性（99.9%）和持久性（99.999999999%）的若干个 *9*。

在本文中，我们认为，共识协议应当承认这一现实：在实际部署中，所有服务器都可能发生故障，而且发生故障的可能性各不相同。因此，没有任何共识协议能够做到 100% 安全或保持活跃。事实上，我们发现，当节点承受 1% 的故障率时，Raft（一种流行的崩溃故障容错协议）在三节点部署中仅有 99.97% 的安全性和活性（§3）。我们认为，共识协议应当提供具体的安全性（相应地，也应提供活性）概率性保证。这一保证应根据具体协议以及服务器特有的故障曲线计算得出。故障曲线刻画了给定服务器独特的、随时间变化的故障特征，并且可以利用现代部署每天持续收集的大量遥测数据进行计算。

此外，采用概率方法推理故障，使我们能够探讨一个引人入胜的机会：利用不同服务器的异构故障曲线，以低得多的金钱或能源成本，提供相同的概率安全性／活性保证！例如，我们发现，可以在九个可靠性较低、故障率为 8% 的节点上运行 Raft，并获得相同的 99.97% 安全性和活性。如果这些资源 10×更便宜（例如，竞价型实例 [17]，较旧的硬件），就能 3× 降低成本。这项分析依赖于将故障曲线纳入现有共识协议。我们认为还可以更进一步，开发概率原生共识协议，利用故障曲线提升性能、降低成本，并绕过传统共识所必需的昂贵法定人数交集不变量。

在本文余下部分，我们首先介绍现实世界故障的特性（§2）。随后，我们将详细分析知名 CFT 和 BFT 协议的概率安全性与活性保证（§3） [18, 61]。最后，我们将探讨：设计真正的原生概率共识协议面临哪些挑战，又有哪些机遇（§4）。

2 故障具有概率性

f 阈值模型为了简化问题而将现实离散化：它假设至多有 f 个节点发生故障，而正确节点永远不会发生故障。它也没有考虑到某些服务器比其他服务器更容易发生故障，因为所有故障都被一视同仁：每个故障都使当前故障计数增加一个单位。我们认为，这种简化适得其反。它掩盖了故障的本质，导致共识协议：1）提供的保证与人们对可靠性的理解相差甚远；2）错失性能优化的机会。相反，服务器故障应建模为概率分布，以反映其固有的异质性和动态特性。我们将其称为服务器的故障曲线 p_u。

大规模硬件故障方面的广泛研究 [25,31, 37, 39, 76] 有助于我们精确刻画 p_u。大型数据中心会详细记录其服务器 [25]或图形处理器 [49]，的观测故障率遥测数据，包括内存故障、位翻转或磁盘驱动器数据块损坏的数量，以及这些情况如何随制造商、写入周期或时间而变化。针对量化现货实例故障率的类似研究也已开展 [4, 43, 77]。除了实验结果，高级分析研究也会预测故障率。例如，高级预测模型可以根据晶体管和芯片老化情况估算故障率 [8–10, 12]。大型组织也会开展大规模威胁与风险分析，以评估机器失陷和攻击的可能性 [70]。

大规模故障研究得出了三个主要结论：（*1*）节点并非同等容易发生故障。大多数共识协议都假设所有节点发生故障的可能性相同。实际上，故障概率很少是均匀的，它取决于硬件制造商、 [39, 48, 64]，硬件在数据中心中的放置位置、 [66]，磁盘容量或使用情况 [64, 66]。故障概率不仅限于机器故障，还可以用来描述人类信任和激励机制等社会概念。在分布式信任的背景下，可以为与之存在长期合同协议的各方指定较低的故障概率 [58]，，而来自敌对国家的成员则可能被指定较高的故障概率。区块链系统中的权益体现了类似的理念：权益较高的节点在系统发生故障时损失更大，因此被认为更加可信 [6, 34]。

（2）故障可能性会随时间演变。当前的共识协议不会对故障可能性进行推理，因此只能通过更改阈值来应对不断变化的故障率f。遗憾的是，更改f十分繁琐，因为这需要付出高昂代价进行重新配置 [27]。然而，故障概率会随时间演变。在软件层面，故障往往会集中出现在重大软件更新前后，CrowdStrike 事故便是如此 [24]，；运营高峰时段和工作负载突然变化时也会出现这种情况。硬件可靠性同样会随时间演变 [25, 39]。例如，磁盘故障符合浴盆曲线：在早期和损耗阶段，发生故障的概率较高，而在有效使用阶段，故障率则相对较低。谷歌和 Facebook 均报告称，随着处理器核心老化，静默损坏错误（处理器计算数据出错）会变得更加常见 [25, 39]。在分布式信任的背景下，故障概率（亦即可信度）可能会随地缘政治背景而变化。

（3）故障具有相关性。协议假设最多存在f个故障，并以统一方式处理故障，这实际上默认了故障彼此独立。遗憾的是，故障往往具有相关性或经过人为策划。在软件层面，故障可能源于定期重启、软件部署或运维更新。在硬件层面，研究表明，以相似方式使用或彼此靠近放置的设备会呈现相似的故障模式。例如，具有相似振动或温度的磁盘会呈现相似的故障模式 [66]。恶意攻击也经常攻陷成批机器。以 Azure Confidential Ledger 或 Signal 的密钥恢复服务等分布式信任场景为例。二者都使用 Intel SGX 或 AMD SEV-SNP 等可信硬件，以增强针对机器失陷的防御能力。当这些架构中发现漏洞时——遗憾的是，这种情况十分常见 [5, 19, 29, 53, 54, 65]，——平台范围攻击的风险就会增加。

（4）不能简单地将故障视为崩溃或拜占庭故障。当前的共识协议（少数例外情况除外） [21] 迫使开发者在两种极端选择中做出其一：要么乐观地假设恶意故障永远不会发生，要么始终承担其发生所带来的成本。实际上，大多数节点会因崩溃而失效，但偶尔也会表现出恶意行为。例如，谷歌和 Facebook 中由易变核心 [25, 39]触发的数据损坏执行错误就属于拜占庭故障。不过，与传统的服务器故障（4% 年故障率）相比，这类故障要罕见得多（谷歌约为 0.01%）。在分布式信任场景中，同样的道理也成立：可信执行环境大多数时候可以阻止拜占庭攻击，但未被诊断出的漏洞仍可能潜伏其中。

与共识协议不同，存储系统的可靠性研究设计了有效的指标来捕捉这些模式。磁盘可靠性以年故障率（AFR）表示，通常在大规模磁盘集群中测量 [48]。存储领域依赖马尔可夫模型来量化平均故障时间（MTTF）、平均故障间隔时间（MTBF）和平均数据丢失时间（MTTDL）等指标 [46, 63]。在马尔可夫模型中，状态表示各种配置（即 运行中的磁盘数量），转移则由磁盘故障、修复或恢复引起，其速率由故障概率（λ）和修复概率（μ）决定。根据稳态概率，MTTDL 和 MTTF 的期望值可指导可靠系统有效机制的设计（例如 在 RAID 中，采用包含 n 块磁盘和 k 块校验磁盘的条带化方案时，其预期 MTTDL）。这些系统可为应用提供可配置的概率保证“九”，符合当今人们对可靠性的认知方式。人们利用这些关于故障的丰富知识，在云 [45, 46]，中部署纠删码数据，从而节省了 40% 以上的磁盘成本。

我们认为，共识领域也可以采取类似的方法。我们设想了两种路径：1）更好地理解并利用现有共识协议所提供的概率保证；2）重新审视共识的核心原语（法定人数交集、领导者选举等）是否可以重新设计，以使用故障曲线和概率保证。下面将依次介绍这两种路径。

3 共识协议分析

我们首先分析现有的共识协议，以了解将故障视为概率事件时，这些协议能提供哪些保证。为简化起见，我们不考虑重新配置（添加或移除节点），并假设故障彼此独立。我们不考虑故障曲线，而是假设每台机器 u 发生故障的概率恒定为 p_u。在这种情况下，机器故障共有 2^N 种可能的组合（故障配置）。每种配置都会产生一组可能的系统运行情况，这些情况可能因消息调度而异。如果某种配置下的所有系统运行情况都能确保未故障节点之间的一致性，我们就称该配置是安全的。如果在所有运行情况下，所有未故障节点最终都提交了所有操作，我们就称该配置具有活性。通过计算每种故障配置发生的可能性，我们可以算出在这一特定部署环境中，某种算法能够保证安全性和活性的总体概率。

3.1 共识入门

大多数共识协议都遵循相似的结构。它们按视图依次推进，每个视图由一位不同的领导者负责，任务是提出客户端操作。在每个视图中，要提交一项操作，就必须经历一系列步骤：在进入下一步之前，一个或多个节点会广播一条消息，并等待收到一组回复（即法定人数）：

步骤 *1*：非互斥性（仅适用于拜占庭容错）。此步骤确保在一个视图中，每个槽位至多只有一项操作能够达成一致。此阶段仅用于防范拜占庭领导者，因为他们可能会向系统中的节点发送相互冲突的提案。参与者获得非互斥性法定人数（Q_{eq}）后，非互斥阶段即告结束。

步骤 *2*：持久性。下一步确保任何（可能）已提交的操作都能在视图和领导者变更期间得到保留。节点获得持久化法定人数（Q_{per}）后，即提交操作。

步骤 *3*：视图变更。当节点检测到共识协议不再取得进展时，会通过获得视图变更法定人数（Q_{vc}）来选出新的领导者。由于虚假视图变更会妨碍活性，大多数拜占庭容错共识协议会确保正确节点只有在从足够多的其他正确节点处获知新视图后，才加入该视图（视图变更触发法定人数 Q_{vc_t}）。

这些法定人数的大小取决于它们希望维持的不变量以及所假设的故障模型。在拜占庭容错中，非互斥性法定人数必须至少相交于一个正确节点，以确保不会针对同一操作形成两个法定人数（正确节点绝不会同时为两者投票）。视图变更法定人数 Q_{vc} 与持久化法定人数 Q_{per} 也必须相交于一个正确节点，从而确保所有已提交操作都会纳入下一个视图。为避免虚假视图变更，必须保证 Q_{vc_t} 至少包含一个正确节点（正确节点不会伪造视图变更）。

在 CFT 场景中，所需的不变量更简单：例如，只需确保跨视图的法定人数至少相交于一个正确节点，即可保证持久性。违反任何一项不变量都会导致安全性违规，而无法形成任何法定人数（例如，由于故障节点过多）则会破坏活性。

3.2 分析与要点

将上述不变量具体应用于 PBFT 和 Raft 这两种常见的拜占庭容错和崩溃容错协议，可针对特定故障配置得出以下两个定理。

定理3.1。
PBFT 安全当且仅当满足以下所有条件：

（*1*） $|Byz| < 2|Q_{eq}| - N$（

2） $|Byz| < |Q_{per}| + |Q_{vc}| - N$

当且仅当以下条件均成立时，*PBFT* 才是活跃的：

(1) $|Byz| \le |Q_{vc_t}| - |Q_{vc}|$*(2)*

$|Correct| \ge |Q_{eq}|, |Q_{per}|, |Q_{vc}|$*(3)*

$|Byz| < |Q_{vc_t}|$

安全条件 (1,2) 规定，对于规模为 N 的系统，法定人数规模必须足够大，以确保至少有一个正确节点

N	$\|Q_{eq}\|$	$\|Q_{per}\|$	$\|Q_{vc}\|$	$\|Q_{vc_t}\|$	安全 %	活跃 %	安全且活跃 %
4	3	3	3	2	99.94%	99.94%	99.94%
5	4	4	4	2	99.9990%	99.90%	99.90%
7	5	5	5	3	99.997%	99.997%	99.997%
8	6	6	6	3	99.99993%	99.995%	99.995%

表1：PBFT 可靠性，均匀 $p_u = 1\%$。

N	$\|Q_{per}\|$	$\|Q_{vc}\|$	S&L $p_u = 1\%$	S&L $p_u = 2\%$	S&L $p_u = 4\%$	S&L $p_u = 8\%$
3	2	2	99.97%	99.88%	99.53%	98.18%
5	3	3	99.9990%	99.992%	99.94%	99.55%
7	4	4	99.99997%	99.9995%	99.992%	99.88%
9	5	5	99.9999998%	99.99996%	99.9988%	99.97%

表 2：均匀节点故障下 Raft 的可靠性 p_u。

。而活性要求法定人数规模足够小，以便始终有足够多的正确节点来组成所述法定人数。f-阈值模型仅通过计算法定人数的规模来判断它们是否相交，但如果我们知道服务器的故障概率，还可以进行更精确的计算。

定理 3.2。
当且仅当以下条件全部满足时，*Raft* 才是安全的：

(1) $N < |Q_{per}| + |Q_{vc}|$ *and*
(2) $N < 2|Q_{vc}|$

Raft 是活跃的，当且仅当：

(1) $|Correct| \ge |Q_{per}|, |Q_{vc}|$

安全条件 (1) 和 (2) 指出，法定人数必须足够大，确保任意两个法定人数至少有一个节点重叠，从而保证 (1) 操作能在不同视图间持续存在，以及 (2) 选出唯一的领导者。

随后，只需将每种安全（或活跃）故障配置的概率相加，就能计算出算法安全和/或活跃的概率。探究故障概率如何影响不同网络和法定人数规模下的安全性/活性保证，可以得到若干有趣的观察结果（表 1 和2）。

无论你是否愿意，共识都具有概率性。 f-阈值协议声称，当 $N = 3$和 $f = 1$时，它们完全安全且活跃。实际上，我们的分析表明，采用$N = 3$ 的 Raft 只有 3 个 9 的安全性和活性（$p_u = 1\%$）（表 2）。

线性规模的法定人数可能过于冗余。PBFT 和 Raft（§3.1）中的法定人数会随网络规模线性增长。例如，Q_{vc_t} 法定人数的规模至少为 $f + 1$（$N = 100$，$|Q_{vc_t}| = 34$），以确保至少有一个正确节点请求进行视图变更 [15]。但这过于冗余：如果 $p_u = 1\%$，则随机选取的五节点法定人数至少包含一个正确节点的概率已经达到十个九，即 99.99999999%。

规模更大的网络即使由可靠性较低的节点组成，也能发挥作用。我们发现，三节点 Raft 集群（$p_u = 1\%$）的安全性/活性概率与包含九个节点的集群相同，后者的可靠性为 $p_u =$

8%。如果可靠性与定价成正比（例如，Spot 实例），就能 3× 降低成本。因此，硬件运营商可以利用这项分析，挑选最可持续、最实惠和/或性能最佳的硬件，而无需在可靠性上做出妥协。运营商也可以重复利用较旧的硬件，以减少碳排放，同时满足可靠性方面的目标服务质量 (QoS) [3, 14, 38, 59, 75]。

Raft 和 PBFT 未能充分利用可靠节点。Raft 和 PBFT 对故障曲线视而不见。考虑一个由 p_u=8% 个节点组成、运行 Raft 的集群。该集群的99。88%安全（表 2）。如果我们用更可靠的节点替换其中三个（p_u=1%）——几乎一半的节点——安全性也只提高到99。98%（表中未显示）。由于 Raft 不知道哪些节点更可靠，它可能只会将数据持久化到不可靠的节点上。如果我们要求法定人数至少包含一个可靠节点（利用对故障曲线的了解），数据持久性将提高到99。994%。

安全性与活性之间存在一种隐蔽但可利用的权衡。共识协议中的 f阈值模型掩盖了安全性与活性之间固有的权衡。揭示这种权衡可以节省资源。考虑 f=1 以及两个 PBFT 系统，一个有 $3f$+1=4 个节点，另一个有 $3f$+2=5 个节点。在 f阈值模型中，两个系统都能容忍1 个故障，因此没有理由部署5 个节点。然而，在概率世界中，我们的分析发现，与 4 个节点相比，使用5 个节点可使 PBFT 的安全性提高 42–60×，而活性仅略有 1.67×下降（表1）——事实上，5 节点系统比 7 节点系统更安全，而后者的部署和运行成本高出 40%。5 节点系统相较于 4 节点系统的安全性提升，源于其法定人数更大。更大的法定人数能提高安全性（交集概率更高），但会降低活性（更少的故障就可能阻碍进展）——在这个例子中，影响很小。尽管 7 节点系统可以多容忍一个故障（f=2），但由于规模更大，发生故障的可能性也随之增加，这在一定程度上抵消了其因能容忍更多故障而获得的安全性提升。f阈值模型不仅掩盖了这些洞见，还会造成误导。

4 概率愿景

我们早期分析所得的启示，为设计概率共识指明了几个富有前景的方向；我们首先概述实现可用的概率共识必须解决的两个挑战：准确刻画故障模型，以及推理端到端保证。

准确的故障曲线。这项研究的关键在于，能否用简单的术语准确表达可能十分复杂的故障曲线。过于悲观的描述会损害性能，而过于简化或乐观的描述则可能导致系统部署后发生故障。故障曲线可以根据遥测数据、用于主动监控故障的机制来计算，也可以通过对硬件故障进行建模的研究来计算，……存储领域早已依靠这类数据对磁盘故障率进行建模 [46]；他们依靠对故障概率以及修复或灾难恢复概率的现实估算，来设计可靠性机制 [63]。随后，这些数值会用于推导平均故障间隔时间（MTBF）或平均数据丢失时间（MTTDL）等指标，从而将可靠性定义为预期的“坏事发生”前的时间。相比之下，共识的设计一向秉持乐观态度，力求防止“坏事”发生。未来的研究应在形式化故障曲线时，解决这种不匹配问题。此外，对相关故障进行建模仍是一项尚未解决的挑战；例如，通常用于计算MTBF和MTTDL的马尔可夫模型，无法捕捉相互依赖的系统状态转移 [37, 74]。

端到端保证。 应用关注端到端可靠性保证，而共识只是整个系统中的一小部分。传统的可靠性保证 [20, 30],以可用性 或持久性的几个九来表示，与共识所提供的概率性安全性和活性并不十分契合。一个可用性为> 0%的共识协议可以确保系统保持活跃。然而，即使共识协议处于活跃状态，如果恢复或重新配置慢得令人无法接受，也可能无法满足可用性要求。除了可用性之外，一个不安全的系统可能会在不同节点提交不同操作，却仍然保持持久，因为两个分叉都得以保留。

迈向概率原生共识。有了准确的故障曲线后，接下来的问题就是如何利用它们。我们的初步分析表明，可以采取多个步骤。首先，我们可以将故障曲线以及概率性安全性/活性纳入现有的 共识协议。例如，我们可以动态选择法定人数规模，使其以很高的概率相互重叠。即便这看似简单的一步也并非易事，因为法定人数并非独立形成，而是必须相交 [41]。这种依赖关系使得计算交集发生的概率变得困难得多，因为切尔诺夫界 [13, 57] 等传统工具已不再适用。

其次，概率方法可以从最可靠的节点中选出领导者，避免选择更容易发生故障的节点。这与 f-阈值模型中的领导者声誉机制 [22, 71] 在理念上相似。这种策略可以改善尾延迟、缩短重新配置延迟，并在节点发生故障时提高安全性。概率方法还可用于设计新型故障检测器 [28, 73], , 使其更加贴近现实且准确。同样，节点可靠性的预测模型能够实现预先重新配置，防止潜在故障危及安全性或活性。

第三，在节点可靠性超过应用需求的部署中，概率协议可以对委员会进行采样，尤其是只选择可靠的节点。

最后，选择完全拥抱共识的概率性本质，使我们能够探索更为激进的设计决策。例如，大多数共识协议都是围绕少数基本概念设计的，例如基于多数派的法定人数交集。概率性抽象要求我们重新构想超越法定人数的共识（例如，如 Ben-Or [16] 或 Rabia [62]）。法定人数系统的本质从定义上就是悲观的：它们保证任意两个法定人数始终相交。然而在实践中，从规模小得多的节点子集中采样，也能以足够高的概率保证交集。同样，强制持久性的法定人数系统过于保守，因为它们考虑的是最坏情况下的对抗场景。理论上，如果任意组合的 $|Q_{per}|$ 个节点发生故障，它们便不再保证安全性。但实际上，$|Q_{per}|$ 次故障导致数据丢失的概率微乎其微。例如，在一个包含 100 个节点的集群中，当 $|Q_{per}| = 10$ 且 $p_u = 10\%$ 时，有50%的概率会发生 $|Q_{per}|$ 次故障。然而，要使这种情况造成数据丢失，$|Q_{per}|$ 次故障必须与最近形成的持久化法定人数完全重叠，其概率为一百亿分之一。

5 相关工作

法定人数系统。先前的工作 [56, 60] 为法定人数引入了负载、容量和可用性的度量；然而，这些工作假设每个节点以相同的概率发生故障。概率仲裁 [7, 13, 57] 放宽了传统的法定人数交集要求，采用规模更小的 $O(\sqrt{N})$ 法定人数，并以高概率相互重叠。

委员会抽样。King 和 Saia [47] 提出了一种共识机制，通过选择网络中的子集，使其中包含的故障服务器比例能够代表整个集群，从而实现 $O(n^{1.5})$ 的通信复杂度。Algo-rand [34]利用可验证随机函数（VRF）实现高效且安全的随机委员会抽样。

细化的故障与信任模型。Upright [21] 为崩溃故障和拜占庭故障引入了各自独立的阈值。基于权益的共识协议 [32, 33, 35] 假定服务器拥有唯一的权益，且超过 f 的权益永远不会同时失效。Stellar [58] 将这一方法加以推广，使对权益分配持有不同视图的服务器之间能够达成集体协议。尽管这些更具表达力的故障模型各自解决了 f 阈值模型的不同缺陷，但它们仍无法充分利用实践中关于故障的丰富知识，以较低开销提供可用的端到端保证。

对 f 阈值系统的分析。Zorfu [11] 使用马尔可夫分析，研究 f 阈值共识系统发生 > f 故障的平均时间。然而，它并未将此分析扩展到平均数据丢失时间 (MTTDL)，也没有根据故障率设计其共识算法。[26, 51] 分析当实际故障数量少于 f 时，f 阈值共识协议的轮次复杂度。[52, 69] 分析实际故障数量超过 f 但少于 $2f$ 时的一致性和问责性。

6 结论

本文认为，f 阈值模型用意良好，但最终并无助益。相反，本文主张应明确刻画硬件故障的概率性与演化特性，正如存储领域已经做到的那样。随着这一转变，我们设想将出现新的、更高效的共识协议，从而更好地契合人们如今对可靠性的思考方式。

参考文献

[1] Amazon S3 中的数据保护。https://docs.aws.amazon.com/AmazonS3/latest/userguide/DataDurability.html。[2] 生产级容器编排 — kubernetes.io。https: //kubernetes.io/。 [访问日期：16-01-2025]。[3] 降低数据中心的碳足迹 — buffalo.edu。https: //www.buffalo.edu/news/news-releases.host.html/content/shared/mgt/news/reducing-carbon-footprint-data-centers.detail.html。[访问日期：13-01-2025]。[4] 竞价型实例中断 - Amazon 弹性计算云 — docs.amazonaws.cn。https://docs.amazonaws.cn/en_us/AWSEC2/latest/UserGuide/spot-interruptions.html。 [访问日期：16-01-2025]。[5] https://community.intel.com/t5/Blogs/Products-and-Solutions/Security/Intel-Google-TDX-Security-Review/post/1471177。[6] 以太坊。https://github.com/ethereum/，2019。[7] Ittai Abraham 和 Dahlia Malkhi。动态系统的概率仲裁。载于《分布式计算：第17届国际会议》，DISC 2003，意大利索伦托，2003年10月1日至3日。会议论文集17，第60–74页。施普林格，2003。[8] Mridul Agarwal、Bipul C Paul、Ming Zhang 和 Subhasish Mitra。电路故障预测及其在晶体管老化中的应用。载于第25届 IEEE VLSI 测试研讨会（VTS'07），第277–286页。IEEE，2007。[9] Hussam Amrouch。提高嵌入式片上系统可靠性的老化、软错误和温度相关技术。博士论文，卡尔斯鲁厄，卡尔斯鲁厄理工学院（KIT），学位论文，2015，2015。[10] Hussam Amrouch、Javier Martin-Martinez、Victor M van Santen、Miquel Moras、Rosana Rodriguez、Montserrat Nafria 和 Jörg Henkel。连接物理层与应用层，以理解老化效应。载于2015年 IEEE 国际可靠性物理研讨会，第3D–1页。IEEE，2015。[11] James W. Anderson、Hein Meling、Alexander Rasmussen、Amin Vahdat 和 Keith Marzullo。强一致性云服务的高可用性本地恢复。IEEE 可靠与安全计算汇刊，14(2):172–184，2017。[12] Md Toufiq Hasan Anik、Sylvain Guilley、Jean-Luc Danger 和 Nagh- meh Karimi。老化对数字传感器的影响。载于2020年第33届国际超大规模集成电路设计会议和2020年第19届国际嵌入式系统会议（VLSID），第189–194页。IEEE，2020。[13] Diogo Avelas、Hasan Heydari、Eduardo Alchieri、Tobias Distler 和 Alysson Bessani。概率性拜占庭容错。载于第43届 ACM 分布式计算原理研讨会论文集，PODC '24，第170–181页，美国纽约州纽约市，2024年。计算机协会。[14] Luiz André Barroso、Urs Hölzle 和 Parthasarathy Ranganathan。《作为计算机的数据中心：仓库规模机器设计》。施普林格

Nature，2019。[15] Mathieu Baudet、Avery Ching、Andrey Chursin、George Danezis、François Garillot、Zekun Li、Dahlia Malkhi、Oded Naor、Dmitri Perel- man 和 Alberto Sonnino。Libra 区块链中的状态机复制。Libra 协会，技术报告，2019。[16] Michael Ben-Or。自由选择的另一项优势（扩展摘要）：完全异步的一致性协议。载于第二届 ACM 分布式计算原理年度研讨会论文集，PODC '83，第27–30页，美国纽约州纽约市，1983年。计算机协会。[17] brittanyrowe。Azure 虚拟机规模集灵活编排模式的 Spot 优先级混合正式可用 — tech- community.microsoft.com。https://techcommunity.microsoft.com/discussions/compute/announcing-general-availability-of-spot-priority-mix-for-azure-virtual-machine-s/3765160，2023。 [访问日期：16-01-2025]。[18] Miguel Castro 和 Barbara Liskov。实用拜占庭容错与主动恢复。ACM 计算机系统汇刊（TOCS），20(4):398–461，2002。[19] Guoxing Chen、Sanchuan Chen、Yuan Xiao、Yinqian Zhang、Zhiqiang Lin 和 Ten H Lai。SGXPectre：通过推测执行从 SGX 飞地窃取英特尔机密信息。载于2019年 IEEE 欧洲安全与隐私研讨会（EuroS&P），第142–157页。IEEE，2019。[20] Asaf Cidon、Stephen Rumble、Ryan Stutsman、Sachin Katti、John Ousterhout 和 Mendel Rosenblum。Copysets：降低云存储中的数据丢失频率。载于2013年 USENIX 年度技术会议（USENIX ATC 13），第37–48页，2013。[21] Allen Clement、Manos Kapritsos、Sangmin Lee、Yang Wang、Lorenzo Alvisi、Mike Dahlin 和 Taylor Riche。Upright 集群服务。载于 ACM SIGOPS 第22届操作系统原理研讨会论文集，SOSP '09，第277–290页，美国纽约州纽约市，2009年。计算机协会。[22] Shir Cohen、Rati Gelashvili、Lefteris Kokoris Kogias、Zekun Li、Dahlia Malkhi、Alberto Sonnino 和 Alexander Spiegelman。了解你的领导者，2021。[23] James C Corbett、Jeffrey Dean、Michael Epstein、Andrew Fikes、Christo- pher Frost、Jeffrey John Furman、Sanjay Ghemawat、Andrey Gubarev、Christopher Heiser、Peter Hochschild 等。Spanner：谷歌的全球分布式数据库。ACM 计算机系统汇刊（TOCS），31(3):1–22，2013。[24] CrowdStrike。技术细节：面向 Windows 主机的 Falcon 更新 | CrowdStrike — crowdstrike.com。https://www.crowdstrike.com/blog/falcon-update-for-windows-hosts-technical-details/。 [访问日期：10- 09-2024]。[25] Harish Dattatraya Dixit、Sneha Pendharkar、Matt Beadon、Chris Mason、Tejasvi Chakravarthy、Bharath Muthiah 和 Sriram Sankar。大规模静默数据损坏。arXiv 预印本 arXiv:2102.11245，2021。[26] Danny Dolev、Ruediger Reischuk 和 H. Raymond Strong。拜占庭一致性中的提前停止。J. ACM，37(4):720–741，1990年10月。[27] Sisi Duan 和 Haibin Zhang。动态拜占庭容错的基础。载于2022年 IEEE 安全与隐私研讨会（SP），第1317–1334页，2022。[28] Dacfey Dzung、Rachid Guerraoui、David Kozhaya 和 Yvonne-Anne Pignolet。永不言弃——概率性与时间故障检测器。载于2016年 IEEE 国际并行与分布式处理研讨会（IPDPS），第679–688页，2016。[29] Shufan Fei、Zheng Yan、Wenxiu Ding 和 Haomeng Xie。SGX 的安全漏洞及对策：综述。ACM 计算调查（CSUR），54(6):1–36，2021。[30] Daniel Ford、François Labelle、Florentina I Popovici、Murray Stokely、Van-Anh Truong、Luiz Barroso、Carrie Grimes 和 Sean Quinlan。全球分布式存储系统中的可用性。载于第9届 USENIX 操作系统设计与实现研讨会（OSDI 10），2010年。[31] Pietro Frigo、Emanuele Vannacc、Hasan Hassan、Victor Van Der Veen、Onur Mutlu、Cristiano Giuffrida、Herbert Bos 和 Kaveh Razavi。Tr- respass：利用目标行刷新机制的多个侧面。载于2020年IEEE安全与隐私研讨会（SP），第747–762页。IEEE，2020年。[32] Chaya Ganesh、Claudio Orlandi 和 Daniel Tschudi。面向隐私感知区块链的权益证明协议。载于《密码学进展——EUROCRYPT 2019：第38届密码技术理论与应用年度国际会议》，德国达姆施塔特，2019年5月19日至23日，论文集，第一部分38，第690–719页。施普林格，2019年。[33] Peter Gaži、Aggelos Kiayias 和 Dionysis Zindros。权益证明侧链。密码学电子印本档案，报告 2018/1239，2018年。https: // eprint.iacr.org/2018/1239。[34] Yossi Gilad、Rotem Hemo、Silvio Micali、Georgios Vlachos 和 Nickolai Zeldovich。Algorand：为加密货币扩展拜占庭协议。载于《第26届操作系统原理研讨会论文集》，SOSP '17，第51–68页，美国纽约州纽约市，2017年。计算机协会。[35] Yossi Gilad、Rotem Hemo、Silvio Micali、Georgios Vlachos 和 Nickolai Zeldovich。Algorand：为加密货币扩展拜占庭协议。载于《第26届操作系统原理研讨会论文集》，第51–68页，2017年。[36] Neil Giridharan、Florian Suri-Payer、Ittai Abraham、Lorenzo Alvisi 和 Natacha Crooks。Autobahn：无缝高速拜占庭容错。载于《ACM SIGOPS第30届操作系统原理研讨会论文集》，第1–23页，2024年。[37] Kevin M Greenan、James S Plank 和 Jay J Wylie。平均失效至无意义时间：{MTTDL}、马尔可夫模型与存储系统可靠性。载于第二届存储与文件系统热点问题研讨会（HotStorage 10），2010年。[38] Udit Gupta、Mariam Elgamal、Gage Hills、Gu-Yeon Wei、Hsien-Hsin S Lee、David Brooks 和 Carole-Jean Wu。ACT：利用体系结构碳排放建模工具设计可持续计算机系统。载于《第49届计算机体系结构年度国际研讨会论文集》，第784–799页，2022年。[39] Peter H Hochschild、Paul Turner、Jeffrey C Mogul、Rama Govindaraju、Parthasarathy Ranganathan、David E Culler 和 Amin Vahdat。无法计数的核心。载于《操作系统热点问题研讨会论文集》，第9–16页，2021年。[40] Heidi Howard、Fritz Alder、Edward Ashton、Amaury Chamayou、Sylvan Clebsch、Manuel Costa、Antoine Delignat-Lavaud、Cédric Fournet、Andrew Jeffery、Matthew Kerner、Fotios Kounelis、Markus A. Kuppe、Julien Maffre、Mark Russinovich 和 Christoph M. Wintersteiger。机密联盟框架：兼具机密性、完整性和高可用性的安全多方应用。VLDB基金会会刊，17(2):225–240，2023年10月。[41] Heidi Howard、Dahlia Malkhi 和 Alexander Spiegelman。灵活Paxos：重新审视法定人数交集。arXiv预印本 arXiv:1608 06696，2016年。。[42] Patrick Hunt、Mahadev Konar、Flavio P Junqueira 和 Benjamin Reed。ZooKeeper：面向互联网规模系统的无等待协调。载于2010年USENIX年度技术会议（USENIX ATC 10），2010年。[43] ju shim。使用 Azure Spot虚拟机 - Azure虚拟机 — learn.microsoft.com。https://learn.microsoft.com/en-us/azure/virtual-machines/spot-vms#eviction-policy。 [访问日期：16-01-2025]。[44] Flavio P. Junqueira、Benjamin C. Reed 和 Marco Serafini。Zab：面向主备系统的高性能广播。载于2011年IEEE/IFIP第41届国际可信赖系统与网络会议（DSN），第245–256页，2011年。[45] Saurabh Kadekodi。磁盘自适应冗余：针对集群存储系统中的磁盘可靠性异构性定制数据冗余。

博士论文，博士论文，卡内基梅隆大学，2020年。[46] Saurabh Kadekodi、K. V. Rashmi 和 Gregory R. Ganger。集群存储系统需要 HeART：利用磁盘可靠性异构性提高存储效率。载于第17届USENIX文件与存储技术会议（FAST 19），第345–358页，美国马萨诸塞州波士顿，2019年2月。USENIX协会。[47] Valerie King 和 Jared Saia。从几乎处处到处处：基于比特的拜占庭一致性。载于《国际分布式计算研讨会》，第464–478页。施普林格，2009年。[48] Andy Klein。Backblaze 2024 年第一季度硬盘统计 — backblaze.com。https: // www.backblaze.com/blog/backblaze-drive-stats-for-q1-2024/，2024年。[访问日期：14-09-2024]。[49] Apostolos Kokolis、Michael Kuchnik、John Hoffman、Adithya Kumar、Parth Malani、Faye Ma、Zachary DeVito、Shubho Sengupta、Kalyan Sal- adi 和 Carole-Jean Wu。重新审视大规模机器学习研究集群的可靠性。arXiv预印本 arXiv:2410.21680，2024年。[50] Leslie Lamport。Paxos 简明教程。ACM SIGACT News（分布式计算专栏）32，4（总第121期，2001年12月），第51–58页，2001年。[51] Christoph Lenzen 和 Sahar Sheikholeslami。一种递归式提前停止阶段 King 协议。载于《2022年ACM分布式计算原理研讨会论文集》，PODC'22，第60–69页，美国纽约州纽约市，2022年。计算机协会。[52] Jinyuan Li 和 David Mazières。超越拜占庭容错系统中的三分之一故障副本。载于第4届USENIX网络系统设计与实现研讨会（NSDI 07），美国马萨诸塞州剑桥，2007年4月。USENIX协会。[53] Mengyuan Li、Luca Wilke、Jan Wichelmann、Thomas Eisenbarth、Radu Teodorescu 和 Yinqian Zhang。系统性研究 AMD SEV-SNP 上的密文侧信道。载于2022年IEEE安全与隐私研讨会（SP），第337–351页。IEEE，2022年。[54] Mengyuan Li、Yinqian Zhang、Huibo Wang、Kang Li 和 Yueqiang Cheng。{CIPHERLEAKS}：通过密文侧信道破解 {AMD}{SEV} 上的常数时间密码学。载于第30届USENIX安全研讨会（USENIX Security 21），第717–732页，2021年。[55] Joshua Lund。安全值恢复技术预览，2019年。[56] Dahlia Malkhi 和 Michael Reiter。拜占庭法定人数系统。载于《第29届ACM年度计算理论研讨会论文集》，STOC '97，第569–578页，美国纽约州纽约市，1997年。计算机协会。[57] Dahlia Malkhi、Michael Reiter 和 Rebecca Wright。概率性法定人数系统。载于《第16届ACM年度分布式计算原理研讨会论文集》，第267–273页，1997年。[58] David Mazieres。Stellar共识协议：互联网级共识的联邦模型。Stellar发展基金会，32 45，2015 1: –。[59] Justin Meza、Qiang Wu、Sanjev Kumar 和 Onur Mutlu。对实际环境中闪存故障的大规模研究。ACM SIGMETRICS性能评估评论，43(1):177–190，2015年。[60] M. Naor 和 A. Wool。法定人数系统的负载、容量与可用性。载于《第35届计算机科学基础年度研讨会论文集》，第214–225页，1994年。[61] Diego Ongaro 和 John Ousterhout。Raft共识算法。计算机科学讲义，190:2022，2015年。[62] Haochen Pan、Jesse Tuglu、Neo Zhou、Tianshu Wang、Yicheng Shen、Xiong Zheng、Joseph Tassarotti、Lewis Tseng 和 Roberto Palmieri。Rabia：通过随机化简化状态机复制。载于《ACM SIGOPS第28届操作系统原理研讨会论文集》，SOSP '21，第472–487页，美国纽约州纽约市，2021年。计算机协会。[63] David A Patterson、Garth Gibson 和 Randy H Katz。廉价磁盘冗余阵列（RAID）的理由。载于1988年 *ACM SIGMOD*国际数据管理会议，第 109–116 页，1988年。[64] Eduardo Pinheiro、Wolf-Dietrich Weber 和 Luiz André Barroso。《大型磁盘驱动器群体中的故障趋势》。发表于第 *5* 届 *USENIX*文件与存储技术会议论文集，FAST '07，第 2 页，美国，2007年。USENIX协会。[65] Pengfei Qiu、Dongsheng Wang、Yongqiang Lyu 和 Gang Qu。《Voltjockey：通过软件控制电压诱发的硬件故障攻破 SGX》。发表于 *2019 AsianHOST*，第 1–6 页。IEEE，2019年。[66] Korakit Seemakhupt、Brent E Stephens、Samira Khan、Sihang Liu、Hassan Wassel、Soheil Hassas Yeganeh、Alex C Snoeren、Arvind Krishnamurthy、David E Culler 和 Henry M Levy。《云规模远程过程调用特征分析》。发表于第 *29* 届操作系统原理研讨会论文集，第 498–514 页，2023年。[67] Alex Shamis、Peter Pietzuch、Burcu Canakci、Miguel Castro、Cedric Fournet、Edward Ashton、Amaury Chamayou、Sylvan Clebsch、Antoine Delignat-Lavaud、Matthew Kerner、Julien Maffre、Olga Vrousgou、Christoph M. Wintersteiger、Manuel Costa 和 Mark Russinovich。《IA-CCF：许可型账本的个人问责》。发表于第 *19* 届 *USENIX*网络系统设计与实现研讨会（*NSDI 22*），第 467–491 页，华盛顿州伦顿，2022年4月。USENIX协会。[68] Yongliang Shen、Kaitao Song、Xu Tan、Dongsheng Li、Weiming Lu 和 Yueting Zhuang。《HuggingGPT：借助 ChatGPT 及其在 Hugging Face 上的伙伴完成 AI 任务》。神经信息处理系统进展，第 36 卷，2024年。[69] Peiyao Sheng、Gerui Wang、Kartik Nayak、Sreeram Kannan 和 Pramod Viswanath。《BFT协议取证》。发表于*2021*年 *ACM SIGSAC*计算机与通信安全会议论文集，CCS '21，第 1722–1743 页，美国纽约州纽约市，2021年。计算机协会。[70] Teodor Sommestad、Mathias Ekstedt 和 Pontus Johnson。《用于安全风险分析的概率关系模型》。计算机与安全，29(6):659–679，2010年9月。[71] Alexander Spiegelman、Balaji Arun、Rati Gelashvili 和 Zekun Li。《Shoal：降低 DAG-BFT 延迟并提高其稳健性》，2023年。[72] Rebecca Taft、Irfan Sharif、Andrei Matei、Nathan VanBenschoten、Jordan Lewis、Tobias Grieger、Kai Niemi、Andy Woods、Anne Birzin、Raphael Poss 等。《CockroachDB：具备韧性的地理分布式 SQL 数据库》。发表于*2020*年 *ACM SIGMOD*国际数据管理会议论文集，第 1493–1509 页，2020年。[73] Jing Tian、Zhi Yang、Wei Chen、Ben Y. Zhao 和 Yafei Dai。《用于高效分布式存储维护的概率故障检测》。发表于*2008*年可靠分布式系统研讨会，第 147–156 页，2008年。[74] Max Tschaikowski 和 Mirco Tribastone。《应对马尔可夫过程代数中的连续状态空间爆炸》。《理论计算机科学》，517:1–33，2014年。[75] Jaylen Wang、Daniel S Berger、Fiodar Kazhamiaka、Celine Irvene、Chaojie Zhang、Esha Choukse、Kali Frost、Rodrigo Fonseca、Brijesh Warrier、Chetan Bansal 等。《面向更低碳排放的云服务器设计》。发表于*2024*年 *ACM/IEEE* 第 *51* 届国际计算机体系结构研讨会（*ISCA*），第 452–470 页。IEEE，2024年。[76] Erci Xu、Mai Zheng、Feng Qin、Yikang Xu 和 Jiesheng Wu。《经验与行动：从 1 万起 {SSD-Related} 存储系统故障中学到了什么》。发表于*2019*年 *USENIX*年度技术会议（*USENIX ATC 19*），第 961–976 页，2019年。[77] Fangkai Yang、Bowen Pang、Jue Zhang、Bo Qiao、Lu Wang、Camille Couturier、Chetan Bansal、Soumya Ram、Si Qin、Zhen Ma 等。《微软云中的竞价型虚拟机驱逐预测》。发表于*2022*年万维网大会伴随论文集，第 152–156 页，2022年。