申请代码	F0203
接收部门	
收件日期	
接收编号	

国家自然科学基金

申 请 书

（2020版）

资助类别： 面上项目

亚类说明：

附注说明：

项目名称： 分布式系统中的因果一致性：规约、协议与验证

申 请 人： 魏恒峰 电 话： 025-83593283

依托单位： 南京大学

通讯地址： 江苏省南京市栖霞区仙林大道163号 南京大学仙林校区

邮政编码： 210023 单位电话： 025-89683827/83593827

电子邮箱： hfwei@nju.edu.cn

申报日期： 2020年01月21日

国家自然科学基金委员会

基本信息

申请人信息	姓　　名	魏恒峰	性别	男	出生年月	1986年09月	民族	汉族
	学　　位	博士	职称	副研究员		每年工作时间（月）		8
	是否在站博士后	否		电子邮箱		hfwei@nju.edu.cn		
	电　　话	025-83593283		国别或地区		中国		
	个人通讯地址	江苏省南京市栖霞区仙林大道163号 南京大学仙林校区						
	工　作　单　位	南京大学/计算机科学与技术系						
	主要研究领域	分布数据一致性、形式化方法						

依托单位信息	名　　称	南京大学		
	联　系　人	朱伟伟	电子邮箱	vivi@nju.edu.cn
	电　　话	025-89683827/83593827	网站地址	scit.nju.edu.cn

合作研究单位信息	单位名称

项目基本信息	项目名称	分布式系统中的因果一致性：规约、协议与验证		
	英文名称	Causal Consistency in Distributed Systems: Specification, Protocols, and Verification		
	资助类别	面上项目	亚类说明	
	附注说明			
	申请代码	F0203.软件理论、软件工程与服务	F0202.系统软件、数据库与工业软件	
	基地类别	计算机软件新技术国家重点实验室		
	研究期限	2021年01月01日 -- 2024年12月31日	研究方向：网构软件	
	申请直接费用	70.0000万元		
中文关键词		网构软件设计方法；开放软件系统模型；因果一致性；形式化规约；形式化验证		
英文关键词		Design methodology for Internetware; Models for open software systems; Causal consistency; Formal specification; Formal verification		

科学问题属性

○ **“鼓励探索，突出原创”**：科学问题源于科研人员的灵感和新思想，且具有鲜明的首创性特征，旨在通过自由探索产出从无到有的原创性成果。

◉ **“聚焦前沿，独辟蹊径”**：科学问题源于世界科技前沿的热点、难点和新兴领域，且具有鲜明的引领性或开创性特征，旨在通过独辟蹊径取得开拓性成果，引领或拓展科学前沿。

○ **“需求牵引，突破瓶颈”**：科学问题源于国家重大需求和经济主战场，且具有鲜明的需求导向、问题导向和目标导向特征，旨在通过解决技术瓶颈背后的核心科学问题，促使基础研究成果走向应用。

○ **“共性导向，交叉融通”**：科学问题源于多学科领域交叉的共性难题，具有鲜明的学科交叉特征，旨在通过交叉研究产出重大科学突破，促进分科知识融通发展为完整的知识体系。

请阐明选择该科学问题属性的理由（800字以内）：

近十年来，因果一致性受到了分布式计算领域与分布式系统领域研究人员的广泛关注。基础理论不断进步，典型系统层出不穷。在规约方面，不同需求催生众多变体；在协议方面，系统模型与实现方式多种多样；在验证方面，前沿工作挑战重重。本项目的目标与价值就是从这三个方面深入研究因果一致性，为相关领域提供理论基础与技术支撑。

在规约方面，我们放宽视野，并不局限在单个因果一致性变体上，而是转而关注各种变体之间的关系，为它们开发有机统一的规约框架。该框架还将有助于以组合的方式揭示新的变体，这些变体的价值将在已有的典型系统中得到论证。

在协议方面，我们在关注最通用的数据分区模型的基础上，着力拓展现有协议的设计空间：一是，从传统的静态系统走向开放的动态系统；二是，考虑协议的容错能力。这两种拓展有助于进一步提升协议的实用性，弥补理论研究与系统研究之间的差距。据我们所知，这方面的相关工作较为匮乏。

在验证方面，我们首次全面考察近年来出现的典型系统，为它们提供严格的规约与正确性证明，填补这些偏系统类工作在理论上的不足。在积累了充足的证明经验之后，我们将研究因果一致性协议的自动验证问题。该问题难度较大，处于相关领域近两三年来的研究前沿。我们的一个长期目标便是为该问题的研究探索思路、引入有效的形式化方法、设计并开发辅助工具等。

中文摘要	为了提高可用性并降低访问延迟，分布式系统通常将数据以副本的形式存放在多个节点上。然而，这也带来了数据一致性问题。近年来，因果一致性受到了广泛关注。理论表明，它是在容忍网络分区条件下，所能实现的最强的数据一致性。本项目将从规约、协议与验证三个方面深入研究因果一致性。在规约方面，我们关注各种因果一致性变体之间的关系，为它们开发统一的规约框架。该框架还将有助于以组合的方式揭示新的变体，这些变体的价值将在已有的典型系统中得到论证。在协议方面，我们着力拓展现有协议的设计空间，进一步提升协议的实用性：一是，从传统的静态系统走向开放的动态系统；二是，考虑协议的容错能力。据我们所知，这方面的相关工作较为匮乏。在验证方面，我们首次全面考察近年来出现的典型系统，为它们提供严格的规约与正确性证明。在此基础上，我们将研究因果一致性协议的自动验证问题。为此，我们计划探索有效的形式化方法、设计并开发辅助证明工具。
英文摘要	For high availability and low latency, distributed systems often replicate data copies on multiple geographically distinct nodes. However, this brings the data consistency problem. Recently causal consistency has attracted widespread attention, which has been proven to be one of the strongest consistency models that are achievable under network partition. In this project, we will study the issues of specification, protocols, and verification of causal consistency. In terms of specification, we will focus on the relationship among variants of causal consistency and develop a unified specification framework for them. This also helps to reveal new variants that will be evaluated in existing causally consistent systems. In terms of protocols, we aim to explore new design spaces, making them more practical. On the one hand, we will turn to dynamic systems from traditional static systems. On the other hand, we will study the resilience of causally consistent protocols. As far as we know, little has been done in these two aspects. In terms of verification of protocols, we will examine various typical systems built in recent years and provide formal specifications and correctness proofs for them. Then, we will study the problem of automatic verification of causally consistent protocols. To this end, we plan to explore effective formal methods and to develop proof assistants.

项目组主要参与者（注：项目组主要参与者不包括项目申请人）

编号	姓名	出生年月	性别	职 称	学 位	单位名称	电话	电子邮箱	证件号码	每年工作时间（月）
1	江雪	1990-12-25	女	博士生	硕士	南京大学	025-83593283	xuejiang1225@gmail.com	3**********0	8
2	黄羿	1993-04-13	男	博士生	硕士	南京大学	025-83593283	yi.huang.njucs@outlook.com	4**********7	8
3	唐瑞泽	1997-02-18	男	博士生	学士	南京大学	025-83593283	DZ1933024@smail.nju.edu.cn	5**********X	8
4	张宇奇	1994-05-03	男	博士生	学士	南京大学	025-83593283	cs_yqzhang@gmail.com	3**********4	6
5	黄开乐	1997-02-21	男	博士生	学士	南京大学	025-83593283	mg1933024@smail.nju.edu.cn	3**********8	7
6	纪业	1995-12-16	男	硕士生	学士	南京大学	025-83593283	jiye@smail.nju.edu.cn	3**********0	6
7	易星辰	1996-12-02	男	硕士生	学士	南京大学	025-83593283	staryi@smail.nju.edu.cn	3**********3	6
8	谷晓松	1997-09-07	男	硕士生	学士	南京大学	025-83593283	xiaosonggu.nju@qq.com	1**********1	6

总人数	高级	中级	初级	博士后	博士生	硕士生
9	1	0	0	0	5	3

国家自然科学基金项目资金预算表（定额补助）

项目申请号： 项目负责人：魏恒峰 金额单位：万元

序号	科目名称	金额
	(1)	(2)
1	**项目直接费用合计**	70.0000
2	1、 设备费	7.4000
3	(1)设备购置费	5.00
4	(2)设备试制费	0.00
5	(3)设备升级改造与租赁费	2.40
6	2、 材料费	6.00
7	3、 测试化验加工费	0.00
8	4、 燃料动力费	4.80
9	5、 差旅/会议/国际合作与交流费	17.00
10	6、 出版/文献/信息传播/知识产权事务费	7.40
11	7、 劳务费	26.40
12	8、 专家咨询费	1.00
13	9、 其他支出	0.00

预算说明书（定额补助）

（请按照《国家自然科学基金项目预算表编制说明》等的有关要求，对各项支出的主要用途和测算理由，以及合作研究外拨资金、单价≥10万元的设备费等内容进行必要说明。）

1. 设备费

设备费预算 7.4 万。本项目需要开发、搭建并测试分布式原型系统，故需购买 2 台笔记本电脑用作客户端，并租赁 3 台阿里云服务器。每台阿里云服务器每月租赁费按 0.1 万元计，4 年内预计共需租赁 8 个月。另外，由于本项目涉及分布式协议自动验证技术，需要运行计算密集型的模型检验与定理证明软件，故需要购置 2 台高性能台式机。具体预算如下：

a) 笔记本电脑 1.5 万/台 × 2 台 = 3.0 万元；

b) 台式机 1.0 万/台 × 2 台 = 2.0 万元；

c) 阿里云服务器 0.3 万/月 × 8 月 = 2.4 万元。

2. 材料费

材料费预算 6.0 万。主要用于购买机器内存、硬盘、U 盘、鼠标、键盘、路由器、硒鼓等计算机耗材、配件及设备维修等费用，按 1.5 万/年计，4 年预计 6.0 万元。

3. 测试化验加工费

无。

4. 燃料动力费

燃料动力费预算为 4.8 万元。课题组使用的服务器机房电费，按服务器机房电费 0.6 万/月，课题组 4 年使用机房时间按 8 个月计算，燃料动力费预算：0.6 万/月 × 8 月 = 4.8 万元。

5. 差旅/会议/国际合作与交流费

该项经费预算为 17.0 万元。主要用于参加在国内召开的学术会议，赴各高校和科研机构学习、交流和调研等学术交流活动，邀请国内外同行专家来访交流，以及赴国外参加学术会议等费用。

测算依据：项目成员共 9 人。a）国内差旅部分，包括参加 NASAC、Internetware、CNCC、SETTA 等国内重要学术会议），4 年共计 15 人次；国内旅费 0.15 万/次，住宿费标准 500 元/天，差旅补助 180 元/天，按 2 天计，平均每次国内差旅费标准 0.25 万/次。b）国际交流与合作部分，赴境外参加国际学术会议，4 年共计 5 人次；国际旅费 0.8 万/次，食宿/公杂费约 250-300 美元/天，签注费/注册费约 0.5 万/次，按 5-7 天计，平均每次国际交流与合作费标准为 2.5 万/次。

a) 国内差旅 15 人次 × 0.3 万/次 = 4.5 万元；

b) 国际合作交流 5 人次 × 2.5 万/次 = 12.5 万元。

6. 出版/文献/信息传播/知识产权事务费

该项经费预算为 7.4 万元。主要用于课题成果发表的论文版面费、专业图书资料购买、网络费和专业通讯费、专利及其它知识产权事务费等。具体预算如下：

a) 论文版面费 4 篇 × 0.5 万/篇 = 2.0 万；（如中国科学，软件学报等刊物）

b) 购买图书资料 4 年 × 0.3 万/年 = 1.2 万；

c) 网络费与专业通讯费 4 年 × 0.4 万/年 = 1.6 万；（如 4G/5G 网络租赁费）

d) 专利与软件著作权费 4 年 × 0.4 万/年 = 1.6 万；

e) 文献查新、资料复印、邮政业务费等 4 年 × 0.25 万/年 = 1.0 万。

7. 劳务费

劳务费预算为 26.4 万。主要用于承担参与课题任务的研究生助研费。助研费标准按 0.12 万/月，8 位研究生预计共投入 220 个工作人月，劳务费预算如下：0.12 万/月 × 220 人月 = 26.4 万元。

8. 专家咨询费

专家咨询费预算为 1.0 万元。主要用于学术研讨会及邀请国内外专家来访的咨询和学术报告费用，按邀请 4 人次左右计算，标准 0.25 万/次，预计专家咨询费支出 1.0 万元。

9. 其他支出

无。

报告正文

参照以下提纲撰写，要求内容翔实、清晰，层次分明，标题突出。**请勿删除或改动下述提纲标题及括号中的文字。**

（一）立项依据与研究内容（建议 8000 字以下）：

1．项目的立项依据（研究意义、国内外研究现状及发展动态分析，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）；

一、研究背景

为了提高可用性并降低访问延迟，分布式系统通常将数据以副本的形式存放在多个节点上[1]。然而，这也带来了数据一致性问题。一方面，强一致性的语义简单、易于推理，但是它会损害系统可用性、且无法容忍网络分区[2][3]。另一方面，最终一致性[4][5]可以保证系统的高性能与高可用性，但是它却使得编程与推理变得复杂[6][7]。

近年来，因果一致性受到了广泛关注。因果一致性要求维护操作之间的因果关系，避免了最终一致性所导致的众多异常。与强一致性相比，因果一致性能够容忍网络分区。事实上，因果一致性是在容忍网络分区条件下，所能实现的最强的数据一致性[8]。**本项目旨在从规约、协议与验证三个方面深入研究分布式系统中的因果一致性。**

规约：因果一致性的核心概念来源于 Leslie Lamport 在分布式消息传递系统中定义的事件间的“先于关系”[9]。Ahamad 等人[10]将先于关系扩展到支持读写操作的多处理器共享内存系统，给出了首个因果一致性定义，称为 CM（Causal Memory）。近年来，在分布式系统背景下，研究人员又提出了因果一致性的多种变体[7][11]。最典型的是，Perrin 等人[11]将定义在读写操作上的因果一致性扩展到支持任意复制数据类型[6]，提出了 WCC（Weak Causal Consistency）与 CCv(Causal Convergence)等变体。这些变体之间有着微妙的差异，对协议设计与验证都会产生影响。为了能够全面深入地研究因果一致性，我们需要为这些变体提供**统一的规约框架**。

协议：与强一致性相比，因果一致性有相对高效的协议实现[10][12][13][14]。然而，这些协议普遍存在两大不足。第一，它们假设系统是

静态的。比如，副本数是固定的或者数据分区是固定的。在实际生产环境中，这种假设可能是不成立的。因此，我们需要研究如何**在动态系统中设计高效的因果一致性**协议。第二，它们没有考虑节点出错的情况。当节点出错时，这些协议可能会违反因果一致性，甚至导致数据丢失。因此，我们需要研究如何**设计具有容错（包括节点崩溃错误与拜占庭错误）能力的因果一致性协议**。

验证：给定协议，如何判断该协议是否实现了它所声称的因果一致性变体？我们将该问题称为因果一致性验证问题。为了解决该问题，我们首先将全面考察近年来出现的典型因果一致性系统[12][13][14][15]。这些系统大多缺少明确的规约，更缺少严格的正确性证明。因此，我们计划为它们提供**严格的规约以及正确性证明**。在此基础上，我们将尝试针对不同的因果一致性变体与协议实现方式总结相应的**证明策略，设计并开发辅助证明工具**。

二、研究意义

本项目旨在从规约、协议与验证三个方面深入研究分布式系统中的因果一致性。在规约方面，为已知的众多因果一致性变体提供统一的规约框架，有助于理清它们之间的关系、甚至揭示更多有用的变体。在协议方面，我们将针对现有协议的不足，设计适用于动态系统的因果一致性协议以及具有容错能力的因果一致性协议。这将有助于扩展因果一致性协议的设计空间。在验证方面，为典型系统提供严格的规约与证明，有助于开发者与使用者更好地理解系统。为因果一致性协议总结证明策略、开发辅助证明工具有助于加深对因果一致性的理解、更好地设计协议并证明它们的正确性。

三、研究现状

下面，我们从规约、协议与验证三个方面论述相关研究工作。

（1）规约：变体众多，缺少框架。

Ahamad 等人[10]在多处理器共享内存系统中定义了因果一致性 CM。CM 的核心在于定义操作之间的因果序。直观地讲，如果两个操作之间存在因果序，那么所有的进程都必须依照因果序执行它们。在此基础上，不同的因果一致性变体有着不同的需求[7][11][16]。Perrin 等人[11]将该定义扩展到任意的复制数据类型，并定义了 WCC 以及 CCv 变体。在 WCC 中，当前操作的返回值不依赖于本地进程之前的操作与其它进程上操作的返回值。在 WCC 的基础上，CCv 要求所有进程按照统一的全序执行所有更新操作，以保证收敛性。Bouajjani 等人[16]给出了违反各类因果一致性变体的“坏模式”。Zennou 等人[17]则基于“坏模式”的思

想进一步定义了 CCM（Convergent CM）与 wCCM（Weak CCM）变体。总而言之，因果一致性存在众多变体，但缺乏统一的规约框架将它们有机地整合在一起。**在本项目中，我们计划对目前常用的一种规约框架进行扩展，使其能涵盖已知的所有因果一致性变体。**

（2）协议：仅针对静态系统；不具有容错能力。

（2.1）动态系统中的因果一致性协议

因果一致性协议的实现方式与系统模型紧密相关[12][18]。针对多处理器共享内存系统，Ahamad 等人[10]设计的因果一致性协议采用简单的 MCS（Memory-Consistency System）模型，它要求进程与数据副本节点一一绑定。该协议假设进程数是固定的，并使用维度等于进程数的向量时钟[19][20]维护操作之间的因果关系。在大规模分布式系统中，进程（或称客户端）数通常远大于副本节点数。在该系统模型下，协议通常使用维度等于副本节点数的向量时钟。为了提高系统的可扩展性，数据分区技术应用普遍。为此，研究人员设计了多种支持数据分区的因果一致性协议。例如，以 GentleRain[13]与 Cure[14]为代表的原型系统采用了“全局稳定”机制。然而，上述协议均假设系统是静态的。如何在动态系统中实现因果一致性尚未得到充分研究[21][22][23][24][25][26]。

Ram 等人[21]研究了如何在移动环境下维护因果一致性。虽然移动环境具有一定的动态性，但它要求环境中存在固定数量的、永不出错的主节点，动态性仅体现在边缘的移动节点上。此外，该协议使用了维度等于对象个数的向量时钟，消息复杂度较高。类似地，Benzaïd 等人[25][26]针对移动环境设计了支持因果序的广播协议。Baldoni 等人[23]针对动态系统设计了同时满足因果一致性与弱持久性的协议。但是，该协议仅实现了单对象因果一致性，没有考虑对象之间的因果关系。此外，它还假设数据副本数是固定的，动态性仅体现在管理这些数据副本的进程可能因失效而被替换。协议所使用的向量时钟的大小是由数据副本数决定的。**在本项目中，我们计划针对动态系统设计满足如下条件的因果一致性协议**：第一，支持对象间的因果一致性，并满足基本的系统活性（如数据持久性）；第二，支持数据副本的动态进入与离开；第三，较低的消息复杂度。

与此相关的研究还包括针对动态系统设计维护线性一致性（Linearizability）[27]的协议[28][29]。根据是否使用了较强的计算原语，我们可以将协议分为两类[30]。以 Rambo[31]为代表的系统在系统重配置阶段使用了共识原语[32]。以 DynaStore[33]为代表的系统则不依赖共识原语，而是通

过较为精巧的算法保证参与者对某种新配置达成一致。我们认为，这两类协议对于在动态系统中设计因果一致性协议都具有重要的参考价值。

（2.2）具有容错能力的因果一致性协议

研究人员通常采用 ALPS 作为衡量因果一致性协议的标准[12]：高可用性（Availability）、低延迟（Latency）、分区容忍性（Partition-tolerance）与高可扩展性（Scalability）。作为首个支持因果一致性的商用数据库，MongoDB 提出了将因果一致性与数据持久性（Durability）相结合的理念[15]。它通过类 Raft[34]协议处理节点失效情况，并采用 Majority Quorum Systems 机制处理操作请求，业务逻辑较为复杂。因此，MongoDB 系统是否能在节点失效时保证数据持久性，仍有待论证。**在本项目中，我们计划深入研究 MongoDB 系统中的数据持久性协议，包括对它进行严格论证。**

最近，Tseng 等人[35]研究了因果一致性协议对节点失效的容错能力。结果表明，如果某因果一致性协议要同时保证基本的活性与收敛性，那么它最多能容忍不超过一半的节点失效。他们还提出了一种具有最优容错能力的因果一致性协议，称为 RCM。但是，RCM 存在两个重要缺点：第一，它使用了客户端向量时钟，空间消耗大且影响系统性能[36]；第二，它采用全副本假设，不支持数据分区，应用场景受限。类似地，Tseng 等人[37]还考察了因果一致性协议对拜占庭故障[38]的容忍能力。结果表明，如果某因果一致性协议要同时保证基本的活性与收敛性，那么它最多能容忍不超过三分之一的节点发生拜占庭故障。他们还提出了一种具有最优拜占庭容错能力的因果一致性协议，称为 Byz-RCM。Byz-RCM 存在与 RCM 相同的缺点。**在本项目中，我们将研究如何克服 RCM 与 Byz-RCM 的缺点，在保证最优容错能力的前提下，设计更具有实用性的高效因果一致性协议。**

（3）验证：缺少严格证明；缺少自动验证技术与辅助证明工具。

据我们所知，大多数系统所实现的因果一致性协议都没有经过严格的证明。这体现在如下三个方面：第一，因果一致性存在多种变体，不少系统并没有明确定义它满足哪一种变体[13][15]；第二，有些系统虽然有明确的规约，但是缺少严格的正确性证明[12]。第三，还有些系统所实现的协议实际上满足比它所声称的规约更强的因果一致性变体[39][40]。**在本项目中，我们计划为近年来出现的典型因果一致性系统提供严格的规约与正确性证明。**

正确性证明的关键在于定义操作之间的因果序，以及为每个客户端构造它应该观察到的操作序列。这与要证明的规约以及协议所采用的系统模型紧密相关。

在规约方面，针对 CM 变体，Ahamad 等人[10]给出了因果一致性协议的首个正确性证明。大量后续相关工作[35][41][42]都采用类似方法证明协议满足 CM。但是，很少有相关工作证明某协议满足其它因果一致性变体。Burckhardt 等人[7]与 Perrin 等人[11]设计并证明了满足 WCCv 的因果一致性协议。然而，这两个协议都是示例性的，缺乏实用性。在系统模型方面，Ahamad 等人[10][41]采用 MCS 模型，客户端与数据副本节点一一绑定。之后，已被证明满足 CM 的协议[35][42]大多采用全副本模型，每个数据副本拥有完整数据集。在这两种系统模型下，可以相对容易地为每个客户端构造合法的操作序列。但是，如何在数据分区模型下为每个客户端构造合法的操作序列，尚未得到充分研究。**在本项目中，我们计划在不同系统模型下，针对不同因果一致性变体，证明典型系统中协议的正确性。**

近年来，有些工作开始研究如何自动验证因果一致性协议的正确性[16][42]。Bouajjani 等人[16]证明了“给定协议是否满足 WCC（CM 或 WCCv）”这一问题是不可判定的。然后，他们将“非数据依赖”类协议[43]的自动验证问题归约到有穷自动机上的可达问题，从而证明了它的可判定性。针对以[10]为代表的一类协议，Lesani 等人[42]提出了能够保障 CM 的“良接收”条件。然而，“良接收”条件并不适用于其它协议实现方式，例如“全局稳定”机制。另外，它也不适用于 CM 之外的变体。据我们所知，在因果一致性协议自动验证方面，相关工作较为匮乏。**在本项目中，我们计划针对各种因果一致性变体以及丰富的实现方式，设计相应的证明策略以及辅助证明工具。**

在一致性协议验证方面，有很多工作关注线性一致性（Linearizability）[27]。例如，工作[44][45]表明在系统节点个数受限的情况下，验证协议是否满足线性一致性是 EXPSPACE-complete 问题。此外，Bouajjani 等人[46]研究了最终一致性协议的自动验证问题。结果表明，该问题可以被归约为可达性与模型检验问题，因而是可判定的。我们认为，这些工作所采用的理论与工具对于研究因果一致性协议验证问题具有重要的借鉴意义。

四、总结

近年来，因果一致性受到分布式计算理论与分布式系统研究人员的广泛关注。**在本项目中，我们计划从规约、协议与验证三个方面深入研究因果一致性。**该项目的主要**目标与价值**在于为因果一致性协议的设计与证明提供理论基础与技术支撑。

参考文献（后文引用也罗列于此）

[1] Bernadette Charron-Bost, Fernando Pedone, and André Schiper (Eds.). 2010. Replication: theory and Practice. *Springer-Verlag*, Berlin, Heidelberg.

[2] Eric A Brewer. Towards robust distributed systems. In *PODC*, page 7, 2000.

[3] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[4] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing update conflicts in bayou, a weakly connected replicated storage system. *SIGOPS Oper. Syst. Rev.*, 29(5): 172–182, Dec. 1995.

[5] Werner Vogels. 2009. Eventually consistent. *Commun. ACM* 52, 1 (January 2009), 40–44.

[6] Sebastian Burckhardt, Alexey Gotsman, Hongseok Yang, and Marek Zawirski. Replicated data types: specification, verification, optimality. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '14, San Diego, CA, USA, January 20-21, 2014, pages 271–284, 2014.

[7] Sebastian Burckhardt. 2014. Principles of Eventual Consistency. *Foundations and Trends in Programming Languages.* 1, 1–2 (October 2014), 1–150.

[8] P. Mahajan, L. Alvisi, and M. Dahlin. Consistency, availability, convergence. *Technical Report TR-11-22*, Computer Science Department, UT Austin, May 2011.

[9] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558 – 565, July 1978.

[10] M. Ahamad , G. Neiger , J. E. Burns et al: Causal Memory: Definitions, Implementation and Programming. *Distributed Computing* (1995) 9: 37.

[11] Matthieu Perrin, Achour Mostefaoui, and Claude Jard. 2016. Causal consistency: beyond memory. In *Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '16). New York, USA, Article 26, 1–12.

[12] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Don't settle for eventual: scalable causal consistency for wide-area storage with COPS. In *SOSP* 2011, 401–416.

[13] J. Du, C. Iorgulescu, A. Roy, and W. Zwaenepoel. Gentlerain: Cheap and scalable causal consistency with physical clocks. In *Proceedings of the ACM Symposium on Cloud Computing*, SOCC 2014, pages 4:1 – 4:13, New York, NY, USA, 2014.

[14] D. D. Akkoorath, A. Z. Tomsic, M. Bravo, Z. Li, T. Crain, A. Bieniusa, N. Preguica, and M. Shapiro. Cure: Strong semantics meets high availability and low latency. In *36th International Conference on Distributed Computing Systems* (ICDCS), 405–414, 2016.

[15] M. Tyulenev, A. Schwerin, A. Kamsky, R. Tan, A. Cabral, and J. Mulrow. Implementation of cluster-wide logical clock and causal consistency in mongodb. In *Proceedings of the International Conference on Management of Data* (SIGMOD), 2019.

[16] Ahmed Bouajjani, Constantin Enea, Rachid Guerraoui, and Jad Hamza. On Verifying Causal Consistency. In *POPL* 2017. ACM, New York, NY, USA, 626–638.

[17] Zennou, Rachid & Bouajjani, Ahmed & Enea, Constantin & Erradi, Mohammed. Gradual Consistency Checking. In *CAV* 2019. New York, NY, USA, 267–285.

[18] Manuel Bravo, Luís Rodrigues, and Peter Van Roy. Saturn: a Distributed Metadata Service for Causal Consistency. In *Proceedings of the Twelfth European Conference on Computer Systems* (EuroSys '17). ACM, New York, NY, USA, 111–126.

[19] C. J. Fidge. Timestamps in Message-Passing Systems That Preserve the Partial Ordering. *Australian Computer Science Communications*, 10(1):56-66, February 1988.

[20] F. Mattern, Virtual Time and Global States of Distributed Systems, *Proc. of the International Workshop on Parallel and Distributed Algorithms*, 1988, pp. 215–226.

[21] D. Janaki Ram, M. Uma Mahesh, N. S. K. Chandra Sekhar, and C. Babu, Causal consistency in mobile environment, *Operating Systems Review* 35 (2001), no. 1, 34–40.

[22] R. Friedman, M. Raynal, and C. Travers, Two abstractions for implementing atomic objects in dynamic systems, In *proceedings of the 9th International Conference on Principles of Distributed Systems* (OPODIS), Pisa, Italy, 2005.

[23] R. Baldoni, M. Malek, A. Milani, and S. Piergiovanni. Weakly-persistent causal objects in dynamic distributed systems. In *25th IEEE Symposium on Reliable Distributed Systems* (SRDS '06), 165–174, 2006.

[24] Alessia Milani. Causal Consistency in Static and Dynamic Distributed Systems. *PhD Thesis* 2006. Dipartimento di Informatica e Sistemistica, Universit`a degli Studi di Roma "La Sapienza", Roma, Italy.

[25] Chafika Benzaid and Nadjib Badache. BMobi_Causal: a causal broadcast protocol in mobile dynamic groups. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing* (PODC '08). ACM, New York, NY, USA, 421.

[26] Chafika Benzaid and Nadjib Badache. An Optimal Causal Broadcast Protocol in Mobile Dynamic Groups. In *2008 International Symposium on Parallel and Distributed Processing with Applications*. 477-484.

[27] M. Herlihy and J. M.Wing. Linearizability: A correctness condition for concurrent objects. *ACM Trans. Program. Lang. Syst*., 12(3):463–492, 1990.

[28] Marcos K. Aguilera, Idit Keidar, Dahlia Malkhi, Jean-Philippe Martin, Alexander Shraer. Reconfiguring Replicated Atomic Storage: A Tutorial. *Bulletin of the EATCS: The Distributed Computing Column*. October 2010.

[29] Peter Musial, Nicolas Nicolaou, and Alexander A. Shvartsman. 2014. Implementing distributed shared memory for dynamic networks. *Commun. ACM*, 57, 6 (June 2014), 88–98.

[30] Leander Jehl and Hein Meling. The Case for Reconfiguration without Consensus: Comparing Algorithms for Atomic Storage. *20th International Conference on Principles of Distributed Systems* (OPODIS 2016), 31:1--31:17.

[31] S. Gilbert, N. Lynch, and A. Shvartsman. Rambo: a robust, reconfigurable atomic memory service for dynamic networks. *Distr. Comp.*, 23(4), 2010.

[32] Leslie Lamport. Paxos made simple. *ACM SIGACT News*, 32(4):18–25, December 2001.

[33] Marcos Kawazoe Aguilera, Idit Keidar, Dahlia Malkhi, and Alexander Shraer. Dynamic atomic storage without consensus. *J. ACM*, 58(2), 2011.

[34] Diego Ongaro and John K Ousterhout. In search of an understandable consensus algorithm. In *USENIX Annual Technical Conference*, pages 305 – 319, 2014.

[35] Lewis Tseng, Zezhi Wang, Yajie Zhao. Resilient distributed causal memory in client-server model. *24th IEEE Pacific Rim International Symposium on Dependable Computing* (PRDC 2019). pages 95—104, 2019.

[36] M. Bravo, N. Diegues, J. Zeng, P. Romano, and L. Rodrigues. On the use of clocks to enforce consistency in the cloud. *IEEE Data Engineering Bulleting*, 38(1):18--31, 2015.

[37] Lewis Tseng, Zezhi Wang, Yajie Zhao, Haochen Pan. Distributed causal memory in the presence of byzantine servers. *18th IEEE International Symposium on Network Computing and Applications* (NCA 2019), pages 1—8.

[38] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem.

ACM Trans. Program. Lang. Syst. 4, 3 (July 1982), 382–401.

[39] K. Petersen, M. J. Spreitzer, D. B. Terry, M. M. Theimer, and A. J. Demers. Flexible update propagation for weakly consistent replication. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, SOSP '97, 288–301, France, 1997.

[40] M. Dahlin, L. Gao, A. Nayate, A. Venkataramana, P. Yalagandula, and J. Zheng. Practi replication. In *Proceedings of the 3rd USENIX Symposium on Networked Systems Design and Implementation*, NSDI '06, 2006.

[41] A. Fern´andez, E. Jim´enez, and V. Cholvi, On the interconnection of causal memory systems. *Journal of Parallel and Distributed Computing* (JPDC 2004), 64(4), 498–506.

[42] M. Lesani, C. J. Bell, and A. Chlipala. Chapar: certified causally consistent distributed key-value stores. In *ACM SIGPLAN Notices*, volume 51, 357–370. ACM, 2016.

[43] P. Wolper. Expressing interesting properties of programs in propositional temporal logic. In *POPL* ’86, 184–193. ACM Press, 1986.

[44] R. Alur, K. L. McMillan, and D. Peled. Model-checking of correctness conditions for concurrent objects. *Inf. Comput.*, 160(1-2), 2000.

[45] J. Hamza. On the complexity of linearizability. In *NETYS* ’15, volume 9466 of Lecture Notes in Computer Science. Springer, 2015.

[46] A. Bouajjani, C. Enea, and J. Hamza. Verifying eventual consistency of optimistic replication systems. In *POPL* ’14, 285–296, New York, NY, USA, 2014. ACM.

[47] Rivka Ladin, Barbara Liskov, Liuba Shrira, and Sanjay Ghemawat. Providing high availability using lazy replication. *ACM Trans. Comput. Syst*., 10(4):360–391, 1992.

[48] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Stronger semantics for low-latency geo-replicated storage. In *NSDI* '13, pages 313--328, 2013.

[49] P. Bailis, A. Ghodsi, J. M. Hellerstein, and I. Stoica. Bolt-on causal consistency. In *SIGMOD* '13, 761--772, New York, New York, USA, 2013.

[50] S. Almeida, J. a. Leitão, and L. Rodrigues. ChainReaction: A causal+ consistent datastore based on chain replication. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys 2013, 85--98, Czech Republic, 2013.

[51] J. Du, S. Elnikety, A. Roy, andW. Zwaenepoel. Orbe: Scalable causal consistency using dependency matrices and physical clocks. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, 11:1--11:14, Santa Clara, California, 2013.

[52] M. Zawirski, N. Preguiça, S. Duarte, A. Bieniusa, V. Balegas, and M. Shapiro. Write fast, read in the past: Causal consistency for client-side applications. In *Proceedings of the 16th Annual Middleware Conference*, Middleware '15, 75--87, BC, Canada, 2015.

[53] M. Ahamad and F.J. Torres-Rojas, Plausible clocks: constant size logical clocks for distributed systems, *Distributed Computing* 12 (1999), 179–195.

[54] Paulo Sérgio Almeida, Carlos Baquero, and Victor Fonte. 2008. Interval Tree Clocks. In *Proceedings of the 12th International Conference on Principles of Distributed Systems* (OPODIS '08). Berlin, Heidelberg, 259–274.

[55] Christian Cachin, Rachid Guerraoui, and Lus Rodrigues. Introduction to Reliable and Secure Distributed Programming (2nd. ed.). Springer Publishing Company, 2011.

2. 项目的研究内容、研究目标，以及拟解决的关键科学问题（此部分为重点阐述内容）；

A. 研究目标

在本项目中，我们计划从规约、协议与验证三个方面深入研究分布式系统中的因果一致性。该项目的主要**目标与价值**在于为因果一致性协议的设计与证明提供理论基础与技术支撑。在**规约**方面，我们计划开发因果一致性规约框架，将现有的各种因果一致性变体有机地组织在一起。在**协议**方面，我们计划探索更广阔的协议设计空间：一是，从传统的静态系统走向动态系统；二是，设计具有容错能力的因果一致性协议。在**验证**方面，我们首先将为近年来出现的典型因果一致性系统提供严格的规约与正确性证明。在此基础上，我们将尝试针对不同变体与不同实现方式总结相应的证明策略，设计并开发辅助证明工具。

B. 研究内容（见表格 1）

（1）因果一致性规约框架

因果一致性在分布式计算理论与分布式系统领域得到了广泛关注。不同的需求催生了因果一致性的众多变体[7][11][16]。为了将这些变体有机地组织在一起，我们需要开发统一的规约框架。该框架不仅可以表达既有的因果一致性变体，还将有助于发现新的变体。为了进一步说明新变体的价值，我们将证明某些典型系统实际上已经实现了该框架所揭示的新变体。

Burckhardt 等人[6][7]基于抽象的可见性（Visibility）关系与仲裁（Arbitration）开发了可以描述任意复制数据类型的一致性规约框架。我们称之为(vis, ar)框架。然而，该框架是专门针对弱一致性模型的，尤其是最终一

表格 1 研究内容、拟解决的关键科学问题以及拟采取的研究方案总览表。

研究主题	研究内容	关键科学问题	研究方案
规约	开发可涵盖已知因果一致性变体的规约框架	如何在(vis,ar)框架基础上进行扩展	分别放松对vis、ar 的要求
	在框架内定义新变体，结合典型系统论证其价值	如何定义变体、选择何种系统	定义 CMv 并考察 MongoDB 系统
协议	针对动态系统设计因果一致性协议	如何定义“动态性”	参考 DynaStore 系统
		设计高效的因果关系维护机制	考察非传统向量时钟
	设计具有容错能力的因果一致性协议	论证 MongoDB 协议的容错能力	分模块证明
		针对数据分区模型，如何在节点失效情况下保障数据持久性	结合 Cure 与 RCM 协议
验证	为典型系统提供严格的规约与正确性证明	如何定义合适的 vis 与 ar 关系	考察各种因果关系维护技术
	开发因果一致性协议自动验证技术与辅助证明工具	如何建模协议、如何描述规约、如何利用形式化方法与技术	使用自动机理论或精化理论

致性。它借鉴并发展了最终一致性的两个重要特点：在“强”的方面，它要求仲裁关系是关于所有操作的全序，用于刻画系统中的冲突消解机制。这对应于最终一致性所要求的收敛性。在“弱”的方面，当解释某个操作的返回值时，它允许忽略（在可见性关系下）之前已发生操作的返回值。这对应于最终一致性对中间状态不加限制的特点。这些使得该框架的表达能力受限。特别地，它无法描述经典的因果一致性 CM。因此，**我们计划扩展上述(vis, ar)框架，使得它能够描述经典的一致性模型，包括已知的所有因果一致性变体。**

（2）因果一致性协议设计

（2.1）动态系统中的因果一致性协议设计

针对现有相关工作的不足，**我们计划针对动态系统设计满足如下条件的因果一致性协议**：第一，支持对象间的因果一致性，并满足基本的系统活性与数据持久性；第二，支持数据副本的动态加入与离开；第三，较低的空间复杂度与消息复杂度。此外，我们希望动态性所带来的影响要尽可能小。也就是说，当没有节点动态加入或离开时，它的性能应与静态系统中的协议表现一致。最后，我们希望能开发原型系统，测试协议在不同程度的动态性下的性能。

（2.2）具有容错能力的因果一致性协议设计

据我们所知，大多数经典的因果一致性协议都不具有容错能力。数据副本节

点失效可能会导致数据丢失。为了防止数据丢失，MongoDB 将因果一致性协议与数据持久性协议组合使用[15]。主节点接收到写操作后，不会立即向客户端返回确认消息。只有当它将写操作成功地传播给它所在分区中超过一半的节点后，它才返回确认消息。这种设计使得 MongoDB 实现的因果一致性协议可以容忍少于一半的节点失效。**在本项目中，我们计划阅读 MongoDB 文档[1]、设计规约[2]以及源代码[3]，整理出它的因果一致性协议与数据持久性协议，并考察它们的正确性。**

MongoDB 的系统模型支持数据分区，但是每个数据分区要求有唯一的主节点，更新操作只能由主节点处理。因此，我们还需要在其它系统模型下研究具有容错能力的因果一致性协议。在介绍相关工作时，我们曾指出虽然 RCM 与 Byz-RCM 协议具有容错能力，但是它们有同样的缺点：不支持数据分区，并使用了开销极大的客户端向量时钟。**在本项目中，我们计划在没有主节点的数据分区模型下设计高效且具有容错能力的因果一致性协议。**

（3）因果一致性验证

（3.1）原型系统中因果一致性协议正确性证明

我们计划为以下原型系统中实现的因果一致性协议提供严格的正确性证明：Bayou[4]、Lazy Replication（简称 LR）[47]、PRACTI[40]、COPS[12]、Eiger[48]、Bolt-on Causal Consistency（简称 BoltOn）[49]、ChainReaction[50]、Orbe[51]、GentleRain[13]、SwiftCloud[52]、Cure[14]与 MongoDB[15]。**我们将从规约框架的角度重新审视这些系统，确定它们所满足的因果一致性变体，并在统一框架下给出严格证明。**在（vis，ar）框架下做证明，关键在于定义合适的 vis 与 ar 关系。我们认为，不同的因果关系维护技术会导致不同的 vis 定义，而不同的系统模型会导致不同的 ar 定义。

在因果关系维护方面，可以将协议分成三类[12][18]。LR 等[47]采用基于日志交换的序列化技术。每个数据中心上执行的所有更新操作都要以某种序列化顺序写入本地日志中。数据中心之间不断地交换日志内容，并按照操作之间的因果序执行由其它数据中心传播过来的更新操作。COPS[12]、Eiger[48]、BoltOn[49]等系统采用显式依赖检测技术。该类协议为每个操作维护它所依赖的操作集合。当接收到由其它数据中心传播过来的更新操作后，数据中心会检测它所依赖的操作集合是否都已在本地执行。只有通过检测，远程更新操作才会被执行。

[1] https://docs.mongodb.com/
[2] https://github.com/mongodb/specifications
[3] https://github.com/mongodb/mongo

Orbe[51]、GentleRain[13]、Cure[14]等系统则采用全局稳定机制。它采用较少的元数据，在因果关系检测方面较为保守，只有当某操作确定全局系统（而不是某个数据分区）进入“稳定”状态后，该操作才能被执行。不同的因果关系维护技术会在操作之间确定不同的可见关系，需要不同的vis定义方式。

在系统模型方面，主要分为MCS模型[10]、全副本模型以及数据分区模型。在MCS模型中，客户端与数据副本节点一一绑定。因此，每个客户端最终都能接收到系统中所有的更新操作。在全副本模型中，每个数据副本节点都拥有完整的数据集。因此，每个数据副本节点（并非每个客户端）最终都能接收到系统中所有的更新操作。在数据分区模型中，每个数据副本节点仅维护部分数据，因此它不会接收到作用在其它分区数据上的更新操作。在（vis，ar）框架下，ar主要用于为更新操作确定顺序。因此，不同的系统模型需要不同的ar定义方式。

（3.2）因果一致性协议自动验证技术及其辅助工具

通过证明一系列典型因果一致性系统的正确性，我们预期可以总结出证明的框架、关键步骤以及vis、ar可选方案等。在此基础上，我们计划探索因果一致性协议的自动验证技术。较为长期的目标是，针对各种变体及其多种实现方式设计相应的证明策略、开发辅助证明工具。

C. 拟解决的关键科学问题（见表格 1）

（1）因果一致性规约框架

Burckhardt 等人[6][7]开发的(vis, ar)框架继承了最终一致性的两大特点。这限制了该框架的表达能力。我们需要从两个方面扩展该框架：一方面，我们需要放松它对收敛性的“强”要求；另一方面，我们需要加强它对中间状态不加限制的“弱”要求。因此，**要解决的关键问题是如何基于(vis, ar)框架进行上述扩展。**此外，为了体现新框架的价值，**我们还需要寻找合适的典型系统，证明它实现了新框架揭示出来的因果一致性新变体。**

（2）因果一致性协议设计

（2.1）动态系统中的因果一致性协议设计

针对动态系统设计因果一致性协议，至少**要解决两个关键问题：**（一）定义合适的系统模型，尤其是精确刻画系统的动态性。很显然，在极端的动态环境下，无法保证因果一致性协议的活性。因此，我们需要在系统模型的实用性与协议设计的可行性之间寻求平衡。更进一步，我们希望研究关于系统动态性的不可能性问题；（二）设计高效的动态因果关系维护机制。由于数据副本节点可以动态加

入与离开，所以不能简单地使用服务器端向量时钟。另外，由于我们需要考虑对象之间的因果关系，所以也不能使用仅针对单个对象的向量时钟[23][24]。

（2.2）具有容错能力的因果一致性协议设计

MongoDB 系统的每个数据分区是一个类 Raft 结构。类 Raft 协议用于控制数据复制过程以及因节点失效引发的选主过程。在 MongoDB 系统中，因果一致性协议与数据持久性协议是和 Raft 协议紧密关联的。因此，**要解决的关键问题是如何在类 Raft 协议框架下，证明 MongoDB 中的因果一致性协议满足数据持久性。**

此外，我们还计划在没有主节点的数据分区模型下设计高效且具有容错能力的因果一致性协议。其中，**要解决两个关键问题**：（一）如何设计高效的、适用于数据分区模型的因果关系维护机制；（二）在没有主节点的情况下，如何设计数据复制机制，防止已被提交的数据因节点失效而丢失。

（3）因果一致性验证

（3.1）原型系统中因果一致性协议正确性证明

在(vis, ar)框架中做证明，最关键的步骤是定义合适的 vis 与 ar 关系。就我们目前的经验而言，**要定义合适的 vis 与 ar 关系，各有一个关键问题需要解决**：（一）我们希望 vis 关系尽可能通用。有些协议并不限于键值对系统，而是适用于任意复制数据类型。针对键值对系统，一种常见的 vis 定义是基于读写操作之间的“写入序”构建起来的。然而，很多复杂的复制数据类型并没有自然的“写入序”。针对这些复制数据类型，我们需要定义更抽象的 vis 关系；（二）在定义 ar 关系时，最大的挑战来自于数据分区模型，更新操作分散在不同的分区中。要解决的关键问题就是如何确定这些更新操作之间的顺序。

（3.2）因果一致性协议自动验证技术及其辅助工具

要自动验证因果一致性协议的正确性，需要解决多个关键问题：（一）如何建模某类协议？（二）采用何种形式化方法描述因果一致性规约？（三）如何将验证问题归约到已知问题或者转化为更易于使用的充分性证明策略。

3. 拟采取的研究方案及可行性分析（包括研究方法、技术路线、实验手段、关键技术等说明）；

A. 拟采取的研究方案（见表格 1）

针对各项研究内容与拟解决的关键科学问题，我们提出如下具体研究方案：

（1）因果一致性规约框架

（vis, ar）框架的两个限制分别与可见性关系 vis 以及仲裁关系 ar 相关。

我们需要分别扩展这两种关系。一方面，为了放松框架对收敛性的要求，可以将仲裁关系放松为偏序关系。另一方面，为了加强框架对中间状态的限制，可以要求在解释某个特定操作的返回值时，不能忽略对其可见的操作的返回值。然而，这样定义的 vis 关系过强，我们希望框架可以允许仅观察到部分指定可见操作的返回值。因此，我们可以将扩展后的框架表达为三元组（vis，ar，V）。其中，vis 含义不变，ar 是定义在所有操作上的偏序关系，V 是操作集。当要解释某个特定操作 o 的返回值时，新框架要求 V 中对 o 可见的操作的返回值不可忽略。为了提高框架的实用性，我们计划分别为 vis、ar 与 V 提供一组常用的候选。它们的一组实例化就对应于某种特定的一致性。特别地，当 vis 包含“会话序”并满足传递性时，我们可以组合不同的 ar 与 V，得到已知的所有因果一致性变体。

除了涵盖已知的经典一致性模型，新框架具有的更大的组合度还有助于发现新的一致性模型。特别地，我们预计可以在新框架内定义两三种新的因果一致性变体。为了说明这些变体的有用性，我们将考察典型的因果一致性系统，分析它们是否满足这些新变体。一种可能的方案是，在新框架中定义 CMv（同时满足 CM 与 Convergence），并论证 MongoDB 系统[15]中的因果一致性协议是否满足 CMv。MongoDB 系统支持数据分区，每个数据分区由一个主节点与多个从节点组成。正是由于主节点的存在，我们猜测该协议可能满足收敛性。然而，由于存在数据分区，各个客户端观察到的操作集并不相同。因此，证明的难点在于如何在数据分区模型下，为所有客户端构造同一个合法的仲裁关系。这需要我们在不断的尝试中总结经验，最终提出可行方案。

（2）因果一致性协议设计

（2.1）动态系统中的因果一致性协议设计

在动态系统中设计因果一致性协议，需要解决系统模型与因果关系维护两个关键问题。在系统模型方面，我们将仔细考察 DynaStore 系统[33]为保障活性对系统动态性所作的限制：直观地讲，在任意时刻，同时离开系统的节点不能超过现有节点总数的一半。也就是说，我们计划使用阈值参数刻画系统的动态性[22]，并从理论上分析不同阈值对因果一致性协议设计的影响。在因果关系维护方面，我们将考察是否可以使用非传统向量时钟。例如，Milani 等人[24]使用了 Plausible Clock[53]，它的优势在于时钟的大小不依赖于节点数目。Almeida 等人[54]定义了 Interval Tree Clocks，其大小随系统中节点的加入与离开动态变化。我们需要将这些向量时钟与系统的其它关键模块（如，可容错的通信模

块）相结合，考察它们的可行性。

（2.2）具有容错能力的因果一致性协议设计

我们需要在类 Raft 协议框架下，证明 MongoDB 中的因果一致性协议满足数据持久性。**我们计划采用分模块证明的方式。**具体而言，我们首先考虑类 Raft 协议的选主过程没有被触发的情况。在这种情况下，可以相对容易地证明因果一致性协议以及数据持久性协议的正确性。然后，我们提取出证明中使用了哪些与选主过程相关的性质。最后，我们只需要证明选主过程确实满足这些性质。

在无主节点的数据分区模型下设计高效且具有容错能力的因果一致性协议，需要有高效的因果关系维护机制以及能够满足持久性的数据复制机制。为此，**我们计划将 Cure 协议[14]中的因果关系维护机制与 RCM 协议[35]中的数据复制机制结合起来。**具体而言，我们计划采用服务器端向量时钟维护操作之间的因果关系，并使用“全局稳定”机制解决多数据分区下操作之间的依赖检测问题。为了保证协议在节点失效情况下仍满足数据持久性，我们计划采用一致广播原语[55]实现数据复制。它可以保证任一副本节点上提交的操作最终都会被所有未失效节点接收并提交，从而防止数据丢失。

（3）因果一致性验证

（3.1）原型系统中因果一致性协议正确性证明

在做证明时，我们要解决的关键问题是针对不同协议定义合适的 vis 与 ar 关系。在定义 vis 方面，针对键值对系统，我们可以将 vis 关系定义为“会话序”与“写入序”的并集的传递闭包[10]。针对更复杂的复制数据类型系统，我们计划从协议所采用的因果关系维护技术入手，分析操作之间的可见性。一种可能的方案是利用协议中的向量时钟定义 vis 关系。在定义 ar 方面，我们需要解决数据分区带来的挑战。一种可能的方案是将 ar 定义为 vis 关系的某种线性拓展，然后根据不同变体对操作返回值的要求进行调整。

（3.2）因果一致性协议自动验证技术及其辅助工具

这是本项目中最具挑战性的部分。我们还没有形成明确的解决方案，但有两种可能的技术值得探索：第一种技术基于自动机与模型检验理论，以 Bouajjani 等人的工作[16]为代表。该工作将一类特定协议的自动验证问题归约为有穷自动机上的可达性问题。第二种技术基于精化理论，以 Lesani 等人的工作[42]为代表。该工作将协议相对于 CM 的自动验证问题转化为构造该协议的具体操作语义与 CM 的抽象操作语义之间的精化关系。为了降低构造难度，该工作进一步提出

了一个充分性证明策略。我们计划在这两份代表工作的基础上考察更多的因果一致性变体与协议实现方式。

B. 可行性分析

从研究背景和研究内容来看，近十年来，因果一致性受到了分布式计算领域与分布式系统领域研究人员的广泛关注。基础理论不断进步，典型系统层出不穷。本项目的主要目标和价值就在于从规约、协议与验证三个重要方面为因果一致性研究提供理论基础与技术支撑。选题切合领域发展，具备基本的可行性基础。

从研究方案来看，针对每一项具体的研究内容，本项目都确定了基本的研究思路或解决方案。具体而言，在规约方面，我们将在已被广泛接受的(vis,ar)框架的基础上进行扩展，开发能够涵盖所有因果一致性变体的新框架。在协议设计方面，我们将考察若干代表性协议所使用的关键技术，在此基础上进行组合与创新。在协议验证方面，虽然难度较大，相关工作较少，我们仍然可以从逐一分析典型系统的正确性入手，不断积累证明方面的经验，总结证明策略，并借助形式化方法领域中的理论知识提升抽象层次，为最终解决自动验证问题奠定基础。

从研究基础来看，申请人从博士阶段开始就一直在从事与分布式数据一致性理论相关的研究，具有良好的理论与技术基础。本项目与申请人目前所主持的国家自然科学基金青年项目在内容上一脉相承，因此具有良好的前期工作基础。

从研究环境来看，本项目依托于南京大学计算机软件研究所与计算机软件新技术国家重点实验室。南京大学计算机软件研究所的学术带头人为吕建院士，多年来得到了国家自然科学基金、国家 973 计划、863 计划等项目的大力支持。申请人所在的研究小组在网构软件方法学、分布式计算与分布式系统、形式化方法等研究方向上的知识积累为本项目提供了坚实的人才基础与技术保障。

综上所述，本项目的选题有领域意识，研究内容明确，研究方案切实可行。

4. 本项目的特色与创新之处；

本项目的特色与创新主要体现在如下三个方面：

（一）在因果一致性规约方面，我们并不局限在单个因果一致性变体上，而是关注各种变体之间的关系，为它们开发统一的规约框架。该框架还将有助于以组合的方式揭示新的变体，这些变体的价值将在已有的典型系统中得到论证。

（二）在因果一致性协议方面，我们着力拓展现有协议的设计空间，进一步提升协议的实用性：一是，从传统的静态系统走向开放的动态系统；二是，考虑协议的容错能力。据我们所知，这方面的相关工作较为匮乏。

（三）在因果一致性验证方面，我们首次全面考察近年来出现的典型系统，为它们提供严格的规约与正确性证明。积累了充足的经验之后，我们将研究因果一致性协议的自动验证问题。该问题难度较大，处于相关领域近两三年来的研究前沿。本项目的一个长期目标便是为该问题的研究探索思路、引入有效的形式化方法、设计并开发辅助工具等。

5. 年度研究计划及预期研究结果（包括拟组织的重要学术交流活动、国际合作与交流计划等）。

本项目研究时间为 4 年。年度研究计划与预期研究成果如下：

A.年度研究计划

（1）2021 年 1 月——2021 年 12 月

研究因果一致性规约框架。开发（vis,ar,V）框架，以组合的方式定义新的因果一致性变体，并结合典型系统论述新变体的价值。预计将深入考察 MongoDB 中的因果一致性协议，论证它是否满足某种新变体。就该研究内容撰写论文。

（2）2022 年 1 月——2022 年 12 月

设计具有容错能力的因果一致性协议。深入研究 GentleRain 与 Cure 协议中用于维护并检测因果关系的“全局稳定”机制。深入研究 RCM 协议中用于保障数据持久性的“一致广播”机制。针对数据分区系统模型，结合上述两种机制，设计尽可能高效的、具有容错能力的因果一致性协议。开发原型系统，测试协议性能。就该研究内容撰写论文。

在此过程中，我们会为典型原型系统 GentleRain 与 Cure 提供严格的规约与正确性证明。证明的关键在于应对“全局稳定”机制对 vis 关系的影响以及数据分区模型对 ar 关系的影响。该证明将为长远的自动验证目标提供经验基础。

（3）2023 年 1 月——2023 年 12 月

针对动态系统设计高效的因果一致性协议。深入研究在动态系统中维护线性一致性的 Rambo 与 Dynastore 协议。重点关注如何严格刻画系统的动态性，思考不同程度的动态性对协议设计可行性的影响。深入研究非传统向量时钟，如 Plausible Clocks 与 Interval Tree Clocks，思考如何将它们于动态系统。就该研究内容撰写论文。

（4）2024 年 1 月——2024 年 12 月

研究因果一致性协议的验证问题。我们将继续推进为一系列典型系统提供严格规约与正确性证明的计划。在此基础上，我们将针对不同因果一致性变体与不

同实现方式总结相应的证明策略，从 vis、ar 关系的角度提炼因果一致性正确性证明的本质，形成证明框架，设计并开发辅助证明工具。就该研究内容撰写论文。

B. 预期研究成果

该项目的主要目标与价值在于为因果一致性协议的规约、设计及证明提供理论基础与技术支撑。预期研究成果主要体现为：

（1）在国内外较高水平学术期刊与学术会议上发表 5-8 篇论文。

（2）开发两个原型系统（预期分别实现“动态系统中的因果一致性协议”与“具有容错能力的因果一致性协议”）。

（3）申请 1 项专利（预期内容为“因果一致性协议自动验证技术与辅助证明工具”）。

（4）支持 1-2 名博士生到国内外高水平实验室长期交流与合作。

此外，在项目研究期间，参与指导博士生 3-5 名，硕士生 4-6 名。项目组成员参加相关领域重要国际会议 DISC/ICDCS/SRDS/OPODIS 等 4-6 次。申请人预计将与西班牙 IMDEA Software Institute 研究人员建立交流合作关系。

（二）研究基础与工作条件

1. 研究基础（与本项目相关的研究工作积累和已取得的研究工作成绩）；

申请人从博士阶段起一直在从事与分布式数据一致性理论相关的研究。申请人于 2016 年 12 月在南京大学取得博士学位。博士论文题目为《分布数据一致性技术研究》，导师为吕建教授与黄宇教授。论文提出了“以应用为导向的”、“多样化、可调节；精细化、可度量”的数据一致性问题研究理念，并以此为指导完成了三份具体工作，分别发表在 CCF-A 类国际期刊 TC、TPDS 与 CCF-B 类国际会议 SRDS 上。申请人获得了 2017 年 CCF 优秀博士学位论文奖。

从 2017 年起，申请人及其研究小组重点关注满足最终一致性的复制数据类型，并以此为主题成功申请了一项国家自然科学基金青年项目“面向分布式系统的复制数据类型理论与技术研究”。截止到目前，我们完成了三份研究工作：（一）针对列表数据结构，我们证明了该领域知名学者 Attiya 等人在 2016 年提出的关于“Jupiter 协议满足弱列表规约”的猜想。这被认为是针对“操作转换”类协议的首个严格证明。该工作以 Brief Announcement 的形式发表于分布式计算领域顶级国际会议 PODC’2018，并以全文形式发表于 OPODIS’2018；（二）在第一份工作的基础上，我们继续深入研究了 Jupiter 协议的各种变体，建立了它们之

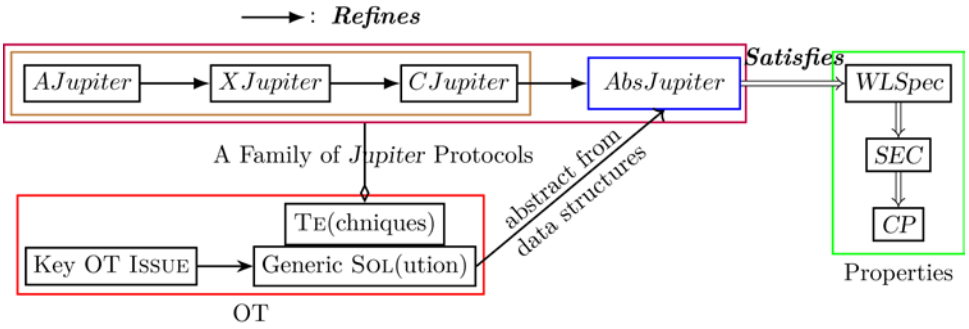

图 1 Jupiter 协议各种变体之间的精化关系与正确性证明框架图。

	k_{max}	$\mathbb{P}(k=1)$	$\mathbb{P}(k=2)$	$\mathbb{P}(k=3)$	$\mathbb{P}(violation)$	Read latency
W2R2	1	100%	0	0	0	100%
W2R1	3	99.9796%	0.0203%	0.0001%	0.0204%	53%
Snitch	2	99.9896%	0.0104%	0	0.0104%	52.3%
Digest	2	99.9926%	0.0074%	0	0.0074%	76.5%
Repair	2	99.9977%	0.0023%	0	0.0023%	53.9%

图 2 数据一致性与访问延迟之间的权衡。

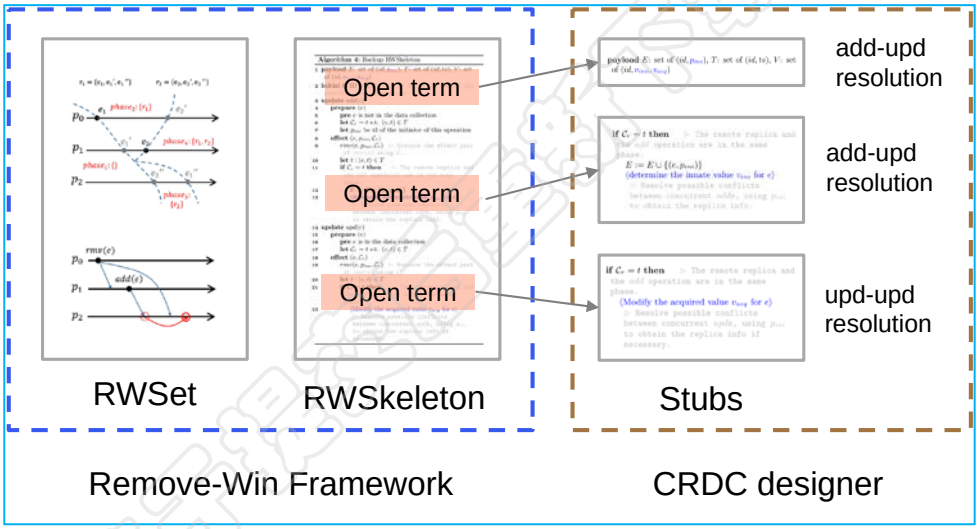

图 3 数据一致性与访问延迟之间的权衡。

间的精化关系，全面系统地证明了它们的正确性（如图 1）。该工作首次厘清了 Jupiter 变体之间的微妙关系，也是首次为各种变体提供全面的正确性证明；

（三）针对读写寄存器，我们在申请人博士论文关于“近乎强一致性”工作的基础上，研究了一组支持多写多读的协议，从概率模型的角度量化分析了它们在数据一致性与访问延迟之间的权衡（如图 2）。后两份工作目前处于审稿期。此外，我们还在尝试为满足最终一致性的无冲突复制数据类型构建协议设计框架(如图 3)。该框架针对一大类复制数据类型实现了“删除优先”(Remove-Win）冲突消解策略，可以极大地简化协议开发人员的设计难度。

本项目计划从规约、协议与验证三个方面深入研究分布式系统中的因果一致

性。这是对博士阶段工作以及青年项目的拓展与深入。对此，申请人及其团队已经具备了良好的理论与技术基础，可以保证项目的顺利开展。

2. 工作条件（包括已具备的实验条件，尚缺少的实验条件和拟解决的途径，包括利用国家实验室、国家重点实验室和部门重点实验室等研究基地的计划与落实情况）；

南京大学计算机软件新技术国家重点实验室目前拥有各类高性能服务器500余台套，其中大型设备公共云计算平台一套（100台多核高性能服务器及55台GPU服务器作为计算资源，同时具备440块Nvidia NVLink V100 GPU卡、460T的高速分布式数据存储能力、IBM虚拟存储网关SVC 2台、30T的IBM Flash System 900闪存阵列2台）、大数据并行计算共享平台一套（140个节点，提供MapReduce、MPI、BSP等多种并行编程模型和环境）、EasyStack云平台一套（包括Dell高性能计算服务器12台）。机房采用160KVA UPS三机系统并机运行，保证各设备24小时不间断工作。万兆接入校园网，同时支持IPv4/IPv6双栈协议，串接万兆安全网关一台，支持防火墙、IPS、WAF等多项安全功能。实验验证、仿真和测试环境非常好。此外，实验室还购买了大量软件资源并订阅了大量的电子刊物全文数据库，例如ACM、Blackwell、Elsevier、IEEE/IEE、Kluwer、Springer、Wiley等，方便研究人员查阅国际最新文献资料。

3. 正在承担的与本项目相关的科研项目情况（申请人和项目组主要参与者正在承担的与本项目相关的科研项目情况，包括国家自然科学基金的项目和国家其他科技计划项目，要注明项目的名称和编号、经费来源、起止年月、与本项目的关系及负责的内容等）；

申请人目前正在主持一项国家自然科学基金青年项目“面向分布式系统的复制数据类型理论与技术研究”。项目编号为61702253，起止年月为2018年01月至2020年12月。申请人全面负责该项目的规划与实施。该项目主要从规约、协议与系统平台的角度研究满足最终一致性的复制数据类型。本项目则从规约、协议与验证的角度研究因果一致性，是青年基金项目内容的自然拓展与深化。

4. 完成国家自然科学基金项目情况（对申请人负责的前一个已结题科学基金项目（项目名称及批准号）完成情况、后续研究进展及与本申请项目的关系加以详细说明。另附该已结题项目研究工作总结摘要（限500字）和相关成果的详细目录）。

无。

（三）其他需要说明的问题

1. 申请人同年申请不同类型的国家自然科学基金项目情况（列明同年申请的其他项目的项目类型、项目名称信息，并说明与本项目之间的区别与联系）。

无。

2. 具有高级专业技术职务（职称）的申请人或者主要参与者是否存在同年申请或者参与申请国家自然科学基金项目的单位不一致的情况；如存在上述情况，列明所涉及人员的姓名，申请或参与申请的其他项目的项目类型、项目名称、单位名称、上述人员在该项目中是申请人还是参与者，并说明单位不一致原因。

无。

3. 具有高级专业技术职务（职称）的申请人或者主要参与者是否存在与正在承担的国家自然科学基金项目的单位不一致的情况；如存在上述情况，列明所涉及人员的姓名，正在承担项目的批准号、项目类型、项目名称、单位名称、起止年月，并说明单位不一致原因。

无。

4. 其他。

无。

魏恒峰 简历

南京大学，计算机科学与技术系，副研究员

教育经历（从大学本科开始，按时间倒序排序；请列出攻读研究生学位阶段导师姓名）：

(1) 2009-9至2016-12，南京大学，计算机软件与理论，博士，导师：吕建

(2) 2005-9至2009-6，南京大学，计算机科学与技术，学士

科研与学术工作经历（按时间倒序排序：如为在站博士后研究人员或曾有博士后研究经历，请列出合作导师姓名）：

(1) 2020-1至现在，南京大学，计算机科学与技术系，副研究员

(2) 2017-1至2019-12，南京大学，计算机科学与技术系，助理研究员

曾使用其他证件信息（申请人应使用唯一身份证件申请项目，曾经使用其他身份证件作为申请人或主要参与者获得过项目资助的，应当在此列明）

主持或参加科研项目（课题）（按时间倒序排序）：

国家自然科学基金委员会，青年科学基金项目，61702253，面向分布式系统的复制数据类型理论与技术研究，2018-01至2020-12，25万元，在研，主持

科学技术部，国家重点研发计划（云计算和大数据专项），2017YFB1001801，可成长的智能化网构软件范型理论、方法与技术研究，2017-10至2021-09，999万元，在研，参加

代表性研究成果和学术奖励情况

（请注意：①投稿阶段的论文不要列出；②对期刊论文：应按照论文发表时作者顺序列出全部作者姓名、论文题目、期刊名称、发表年代、卷（期）及起止页码（摘要论文请加以说明）；③对会议论文：应按照论文发表时作者顺序列出全部作者姓名、论文题目、会议名称（或会议论文集名称及起止页码）、会议地址、会议时间；④应在论文作者姓名后注明第一/通讯作者情况：所有共同第一作者均加注上标“#”字样，通讯作者及共同通讯作者均加注上标“*”字样，唯一第一作者且非通讯作者无需加注；⑤所有代表性研究成果和学术奖励中本人姓名加粗显示。）

按照以下顺序列出：

一、代表性论著（包括论文与专著，合计5项以内）

(1) **Hengfeng Wei**; Yu Huang*; Jian Lu; Specification and Implementation of Replicated List: The Jupiter Protocol Revisited, *The 22nd International Conference on Principles of Distributed Systems (OPODIS)*, Hong Kong, China, 2018-12-17至2018-12-19. **（会议论文）**

(2) **Hengfeng Wei**; Yu Huang*; Jian Lu; Brief Announcement: Specification and Implementation of Replicated List: The Jupiter Protocol Revisited, *The 37th ACM Symposium on Principles of Distributed Computing (PODC)*, Egham, United Kingdom, 2018-7-23至2018-7-27. **(会议论文)**

(3) **Hengfeng Wei**; Yu Huang*; Jian Lu; Parameterized and Runtime-tunable Snapshot Isolation in Distributed Transactional Key-value Stores, *International Symposium on Reliable Distributed Systems (SRDS)*, Hong Kong, China, 2017-9-26至2017-9-29. **(会议论文)**

(4) **Wei, Hengfeng**; Huang, Yu*; Lu, Jian; Probabilistically-Atomic 2-Atomicity: Enabling Almost Strong Consistency in Distributed Storage Systems, *IEEE Transactions on Computers (TC)*, 2017, 66(3): 502-514. **(期刊论文)**

(5) **Wei, Hengfeng**; De Biasi, Marzio; Huang, Yu*; Cao, Jiannong; Lu, Jian; Verifying Pipelined-RAM Consistency over Read/Write Traces of Data Replicas, *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2016, 27(5): 1511-1523. **(期刊论文)**

附件信息

序号	附件名称	备注	附件类型
1	OPODIS2018-Jupiter-Wei		代表性论著
2	PODC-BA2018-Jupiter-Wei	Brief Announcement	代表性论著
3	SRDS2017-RVSI-Wei		代表性论著
4	TC2017-PA2AM-Wei		代表性论著
5	TPDS2016-VPC-Wei		代表性论著

（草稿）请于提交后重新下载PDF的正式稿

项目名称： 分布式系统中的因果一致性：规约、协议与验证

资助类型： 面上项目

申请代码： F0203.软件理论、软件工程与服务

国家自然科学基金项目申请人和参与者科研诚信承诺书

本人**在此郑重承诺**：严格遵守中共中央办公厅、国务院办公厅《关于进一步加强科研诚信建设的若干意见》规定，所申报材料和相关内容真实有效，不存在违背科研诚信要求的行为；在国家自然科学基金项目申请、评审和执行全过程中，恪守职业规范和科学道德，遵守评审规则和工作纪律，杜绝以下行为：

（一）抄袭、剽窃他人科研成果或者伪造、篡改研究数据、研究结论；

（二）购买、代写、代投论文，虚构同行评议专家及评议意见；

（三）违反论文署名规范，擅自标注或虚假标注获得科技计划等资助；

（四）购买、代写申请书；弄虚作假，骗取科技计划项目、科研经费以及奖励、荣誉等；

（五）在项目申请书中以高指标通过评审，在项目计划书中故意篡改降低相应指标；

（六）以任何形式探听尚未公布的评审专家名单及其他评审过程中的保密信息；

（七）本人或委托他人通过各种方式及各种途径联系有关专家进行请托、游说，违规到评审会议驻地游说评审专家和工作人员、询问评审或尚未正式向社会公布的信息等干扰评审或可能影响评审公正性的活动；

（八）向评审工作人员、评审专家等提供任何形式的礼品、礼金、有价证券、支付凭证、商业预付卡、电子红包，或提供宴请、旅游、娱乐健身等任何可能影响评审公正性的活动；

（九）其他违反财经纪律和相关管理规定的行为。

如违背上述承诺，本人愿接受国家自然科学基金委员会和相关部门做出的各项处理决定，包括但不限于撤销科学基金资助项目，追回项目资助经费，向社会通报违规情况，取消一定期限国家自然科学基金项目申请资格，记入科研诚信严重失信行为数据库以及接受相应的党纪政纪处理等。

申请人签字：

编号	参与者姓名 / 工作单位名称（应与加盖公章一致）/ 证件号码	签字
1	**江雪** / 南京大学 / 3**********0	
2	**黄羿** / 南京大学 / 4**********7	
3	**唐瑞泽** / 南京大学 / 5**********X	
4	**张宇奇** / 南京大学 / 3**********4	
5	**黄开乐** / 南京大学 / 3**********8	
6	**纪业** / 南京大学 / 3**********0	
7	**易星辰** / 南京大学 / 3**********3	
8	**谷晓松** / 南京大学 / 1**********1	
9		

项目名称： 分布式系统中的因果一致性：规约、协议与验证
资助类型： 面上项目
申请代码： F0203.软件理论、软件工程与服务

国家自然科学基金项目申请单位科研诚信承诺书

本单位依据国家自然科学基金项目指南的要求，严格履行法人负责制，**在此郑重承诺：**本单位已就所申请材料内容的真实性和完整性进行审核，不存在违背中共中央办公厅、国务院办公厅《关于进一步加强科研诚信建设的若干意见》规定和其他科研诚信要求的行为，申请材料符合《中华人民共和国保守国家秘密法》和《科学技术保密规定》等相关法律法规，在项目申请和评审活动全过程中，遵守有关评审规则和工作纪律，杜绝以下行为：

（一）采取贿赂或变相贿赂、造假、剽窃、故意重复申报等不正当手段获取国家自然科学基金项目申请资格；

（二）以任何形式探听未公开的项目评审信息、评审专家信息及其他评审过程中的保密信息，干扰评审专家的评审工作；

（三）组织或协助项目团队向评审工作人员、评审专家等提供任何形式的礼品、礼金、有价证券、支付凭证、商业预付卡、电子红包等；宴请评审组织者、评审专家，或向评审组织者、评审专家提供旅游、娱乐健身等任何可能影响科学基金评审公正性的活动；

（四）包庇、纵容项目团队虚假申报项目，甚至骗取国家自然科学基金项目；

（五）包庇、纵容项目团队，甚至帮助项目团队采取“打招呼”等方式，影响科学基金项目评审的公正性；

（六）在申请书中以高指标通过评审，在计划书中故意篡改降低相应指标；

（七）其他违反财经纪律和相关管理规定的行为。

如违背上述承诺，本单位愿接受国家自然科学基金委员会和相关部门做出的各项处理决定，包括但不限于停拨或核减经费，追回项目经费，取消一定期限国家自然科学基金项目申请资格，记入科研诚信严重失信行为数据库以及主要责任人接受相应党纪政纪处理等。

依托单位公章:
日期： 年 月 日

合作研究单位公章:
日期： 年 月 日

合作研究单位公章:
日期： 年 月 日