软件学报ISSN 1000-9825, CODEN RUXUEW E-mail: jos@iscas.ac.cn

*Journal of Software*, [doi: 10.13328/j.cnki.jos.000000] http://www.jos.org.cn

©中国科学院软件研究所版权所有. Tel: +86-10-62562563

分布式系统中的因果一致性综述[[1]](#footnote-1)\*

江雪1, 魏恒峰1,2, 黄宇1

1(南京大学 计算机科学与技术系, 江苏 南京 210023)

2(南京大学 软件学院, 江苏 南京 210023)

通讯作者: 魏恒峰, E-mail: hfwei@nju.edu.cn

摘 要:

关键词: 因果一致性;协议;理论;检测

中图法分类号: TP311

中文引用格式: 江雪,魏恒峰,黄宇.分布式系统中的因果一致性综述.软件学报,2023,32(7). http://www.jos.org.cn/1000-9825/0000.htm

英文引用格式: Jiang X, Wei HF, Huang Y. Survey on causal consistency. Ruan Jian Xue Bao/Journal of Software, 2023 (in Chinese).http://www.jos.org.cn/1000-9825/0000.htm

Survey on Causal Consistency in Distributed Systems

JIANG Xue1, WEI Heng-Feng1,2, HUANG Yu1

1(Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

2(School of Software, Nanjing University, Nanjing 210023, China)

**Abstract**:

**Key words**: causal consistency; protocol; theory; testing

# 引言

使用副本的分布式系统

CAP, PACELC, 弱一致性

因果一致性,非事务和事务

规约,理论,协议,检测几个部分的重要性,每个部分大致包括哪些内容

# 规约框架

在本章中我们将介绍针对弱一致复制数据类型的通用规约框架.该框架是通过扩展 Burckhardt 等人[1,2]提出的针对最终一致的复制数据类型的规约框架获得的.

## 关系和序

一个二元关系

# 非事务因果一致性

因果一致性是在分布式系统中被广泛使用的一种弱一致性模型[3,4].因果一致性的规约中最关键的概念是事件之间的发生顺序(happens-before order/hb)[2,5].直观地说,因果一致性确保如果一个事件发生在事件之前,那么所有会话(session)必须在之前观察到.然而,同时并发事件可能被不同的会话以不同的顺序观察到.

相关文献中定义了几种因果一致性的变体,他们之间有细微的差别[2,6,7].在本节中,我们考虑六种变体,称之为Weak Causal Consistency(WCC)、Weak Causal Convergence(WCCv)、Causal Memory(CM)、Causal Memory Convergence(CMv)、Strong Causal Consistency(SCC)和Strong Causal Convergence(SCCv),如表2所定义.请注意,这些变体在相关工作中可能有不同的名称,如表2所总结的.

就可见性()而言,所有因果一致性变体需要来捕捉事件之间发生的先后关系.在仲裁()方面,它们都要求,用以强制要求当解释事件在其操作上下文中的返回值时,可见的所有事件()执行hb顺序.然而,它们可能在以下两个方面有所不同:

* 用于指定其返回值必须被考虑的可见事件子集的函数有多大?请注意,在因果一致性模型中,对于任何事件(因为).因此,候选.
* 仲裁关系对于解决冲突有多强?给定,根据是否是全序,我们区分两种情况.

表 1 因果一致性变体在((框架中的规约

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Consistency Models | Alternative Names |  |  |  |
| WCC ([6]) | CC ([7]) |  |  |  |
| CM ([3,7]) | CC ([6]) |  |
| SCC |  |  |
| WCCv | CCv ([6,7])  CAUSALCONSISTENCY ([2])  CAUSALITY ([8]) |  |  |
| CMv |  |  |
| SCCv |  |  |

## 规约

### Weak Causal Consistency

在[6]中提出的Weak Causal Consistency(WCC)是我们考虑的最弱的因果一致性变体.非正式地讲,一个抽象执行满足WCC,当且仅当每个事件的返回值)能够被其可见事件集的某种序列化所解释(忽略所有可见事件的返回值).更具体地说,在仲裁方面,WCC对收敛性不做额外的限制,因此并发的事件可能被不同的会话以不同的顺序观察到.就而言,WCC允许每个事件忽略其可见事件的所有返回值.

其可见事件的所有返回值.

1. (Weak Causal Consistency)

### Causal Memory

Causal Memory(CM)最初是由Ahamad等人[3]提出的针对读/写寄存器(read/write register)的定义。最近，Perrin等人[6]将其扩展到任意的复制数据类型.CM比WCC更强,因为在证明每个事件的返回值)的合理性时, CM不仅考虑到了WCC中对可见的事件集的操作调用,还考虑到同一会话中之前的事件集合中事件的返回值.换句话说,与WCC相比,CM要求每个会话与之前提供的返回值相一致[6]。

1. (Causal Memory)

### Strong Causal Consistency

我们通过进一步要求每个会话和其他会话提供的返回值保持一致,来加强CM使其成为 SCC(Strong Causal Consistency).形式上，我们有.

1. (Strong Causal Consistency)

### Weak Causal Convergence

WCCv(Weak Causal Convergence)[2,6]是WCC的收敛版本.它通过对执行中的所有事件施加一个全序加强WCC,这为所有会话提供了一个解决并发事件引起的冲突的统一方法. 因此，每个事件的返回值在对可见的事件的集合上进行验证时(忽略他们所有的返回值),这些可见事件要按照共同的全序排序.

1. (Weak Causal Convergence)

### Causal Memory Convergence(CMv)

CMv(Causal Memory Convergence)是CM对应的收敛版本.它要求是一个全序.

1. (Causal Memory Convergence)

### Strong Causal Convergence(SCCv)

SCCv(Strong Causal Convergence)是SCC对应的收敛版本.它要求是一个全序.

1. ()Strong Causal Convergence)

## 理论

## 协议

系统类文章的协议的正确性证明

## 检测

# 事务因果一致性

## 规约

## 理论

## 协议

系统类文章的协议的正确性证明

考虑抽象类似的几个协议；参考 chapar 的证明

## 检测

# 总结与展望

References:

[1] Burckhardt S, Gotsman A, Yang H, Zawirski M. Replicated data types: specification, verification, optimality. Proceedings of the 41st ACM Symposium on Principles of Programming Languages (POPL’2014). : 271–284.

[2] Burckhardt S. Principles of eventual consistency. Found. Trends Program. Lang., Now Publishers Inc., 2014, 1(1–2): 1–150.

[3] Ahamad M, Neiger G, Burns JE, Kohli P, Hutto PW. Causal memory: definitions, implementation, and programming. Distributed Computing, 1995, 9(1): 37–49.

[4] Lloyd W, Freedman MJ, Kaminsky M, Andersen DG. Don’t settle for eventual: scalable causal consistency for wide-area storage with cops. Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP’2011). : 401–416.

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

[6] Perrin M, Mostefaoui A, Jard C. Causal consistency: beyond memory. Proceedings of the 21st ACM Symposium on Principles and Practice of Parallel Programming (PPoPP’2016). .

[7] Bouajjani A, Enea C, Guerraoui R, Hamza J. On verifying causal consistency. Proceedings of the 44th ACM Symposium on Principles of Programming Languages (POPL’2017). : 626–638.

[8] Viotti P, Vukolić M. Consistency in non-transactional distributed storage systems. ACM Comput. Surv., 2016, 49(1): 19:1-19:34.

附中文参考文献:

[2] 陈翔,顾庆,刘望舒,刘树龙,倪超.静态软件缺陷预测方法研究.软件学报,2016,27(1):1−25. http://www.jos.org.cn/1000-9825/4923. htm [doi: 10.13328/j.cnki.jos.004923]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 江雪(1990－),女,博士生,主要研究领域为分布式计算和形式化方法. |  |  | 黄宇(1981－),男,博士,教授,CCF专业会员,主要研究领域为分布式算法和形式化方法. |
|  | 魏恒峰(1986－),男,博士,主要研究领域为分布式计算(尤其是分布式数据一致性 问题(和形式化方法. |  |  |  |

1. \* 基金项目: 国家自然科学基金(61906090, U20B2064, 61773208); 江苏省自然科学基金(BK20191287, BK20170809); 中央高校基本科研业务费专项资金(30920021131); 中国博士后科学基金(2018M632304)

   收稿时间: 2020-04-13; 修改时间: 2020-10-26; 采用时间: 2020-12-14 [↑](#footnote-ref-1)