

分布数据一致性理论与技术研究

魏恒峰

导师: 吕建 黄宇

南京大学软件所

2016 年 7 月 25 日

分布数据一致性理论与技术研究

1. 研究背景: 分布数据
2. 研究问题: 数据一致性
3. 研究方法: 理论模型 + 技术框架
4. 主要工作: VPC + PA2AM + RVSI
5. 未来工作

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

5 未来工作

分布式应用



新浪微博社交网站¹:

- ▶ 日均用户近一亿
- ▶ 日均消息近一亿条

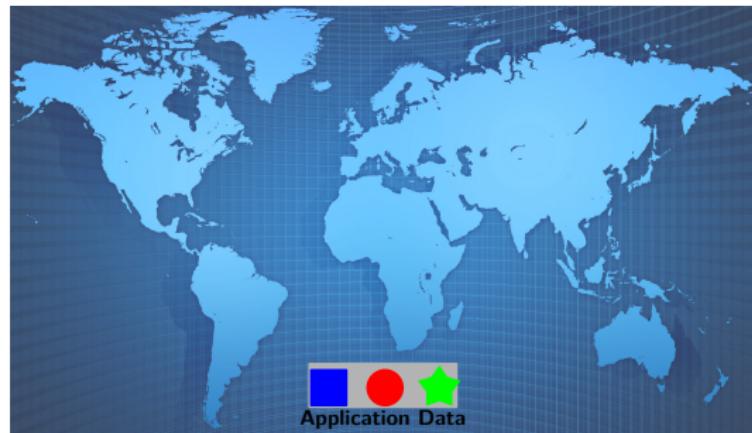
底层数据服务系统特性需求 (H^3L):

- ▶ 低延迟, 高可用性 (4 个 9²)
- ▶ 高容错性, 高可扩展性

¹ 2015 第三季度; 数据来自 China Internet Watch.

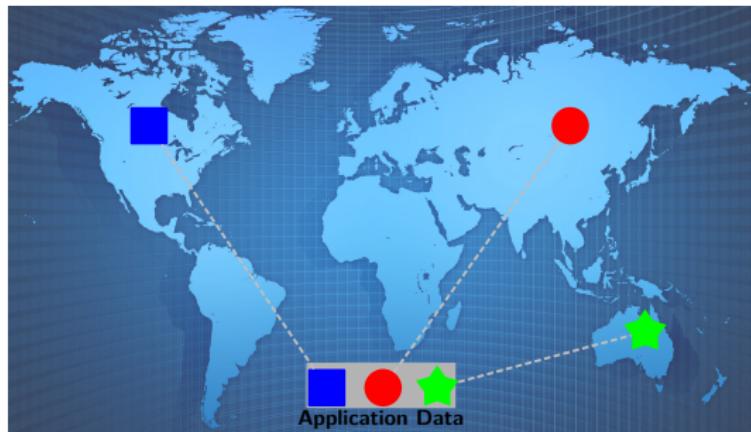
² 数据来自 InfoQ.

分布数据



应用数据:

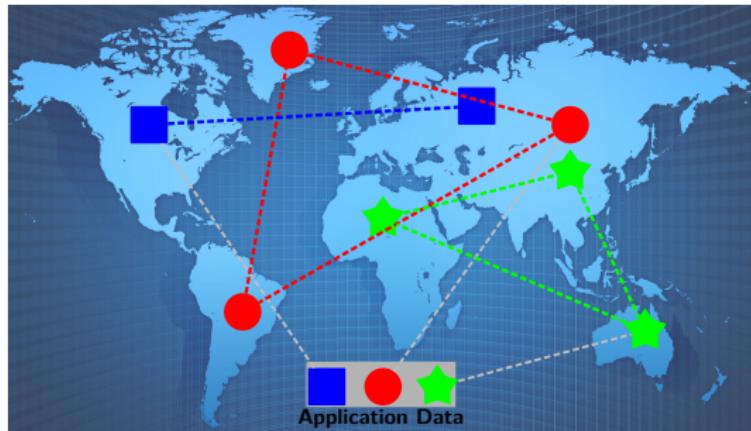
分布数据



应用数据:

1. 分区 (*partition*): 水平扩展

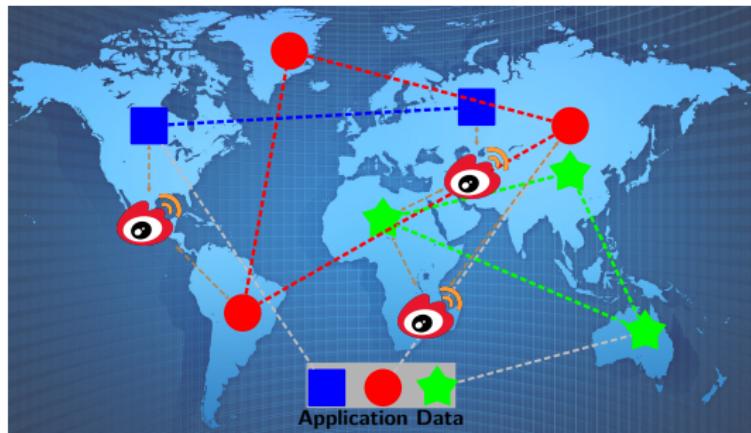
分布数据



应用数据:

1. 分区 (*partition*): 水平扩展
2. 副本 (*replication*): 就近访问, 容灾备份

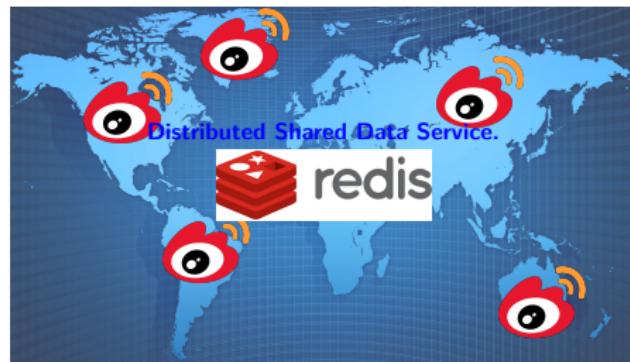
分布数据



分布数据 (distributed data):

1. 分区 (partition): 水平扩展
2. 副本 (replication): 就近访问, 容灾备份

分布共享数据服务



分布共享数据服务 (中间件):
(Distributed Shared Data Service)

屏蔽底层数据分布性 提供共享数据抽象 简化上层应用开发

分布共享数据服务典型应用 (I)



图: 分布式存储系统 (开源 [左] & 商用 [右]).

分布共享数据服务典型应用 (II)

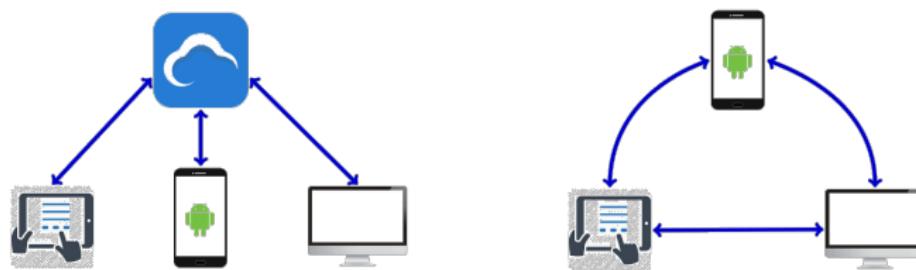


图: 个人多设备文件共享 ([基于云] C/S 结构 [左] & P2P 结构 [右]).

功能需求: 文件副本 [Strauss@MIT Thesis'10]

网络断连: 备份容灾; 离线可用

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

5 未来工作

分布数据一致性问题

理想情况：

- ▶ one-size-fits-all 一致性模型
- ▶ 始终观察到最新副本

没有分布数据一致性问题

分布数据一致性问题

理想情况:

- ▶ one-size-fits-all 一致性模型
- ▶ 始终观察到最新副本

没有分布数据一致性问题

实际情况 (tradeoffs):

H^3L
Partition-tolerance
Convergence
Churn
...
Consistency



分布数据一致性问题

理想情况:

- ▶ one-size-fits-all 一致性模型
- ▶ 始终观察到最新副本

~~没有分布数据一致性问题~~

实际情况 (tradeoffs):

H^3L
Partition-tolerance
Convergence
Churn
...

Consistency



分布数据一致性是分布共享数据服务的核心、挑战性问题

数据一致性问题研究的历史阶段

关于分布数据一致性问题:

基本观点: 传统问题; 新平台带来新挑战

我们的工作: 总结并应对挑战

数据一致性问题研究的历史阶段

关于分布数据一致性问题:

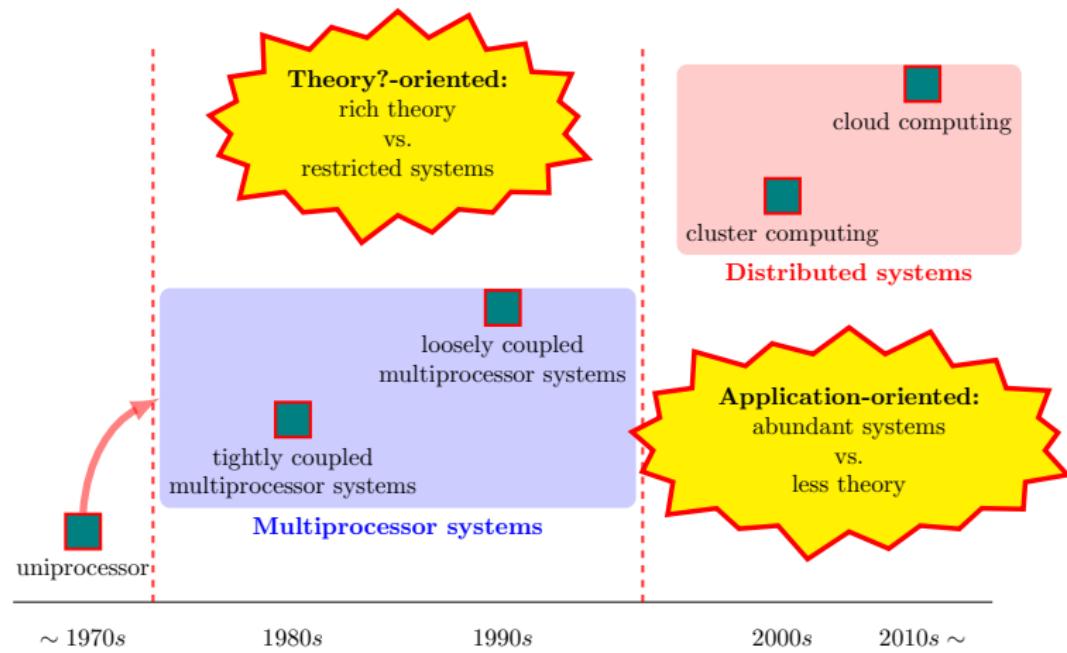
基本观点: 传统问题; 新平台带来新挑战

我们的工作: 总结并应对挑战

从“两个方面”考察研究的历史阶段:

1. 理论 vs. 系统
2. 以数据一致性为核心的 tradeoffs

数据一致性问题研究的历史阶段



数据一致性问题研究的历史阶段 (多处理器系统)

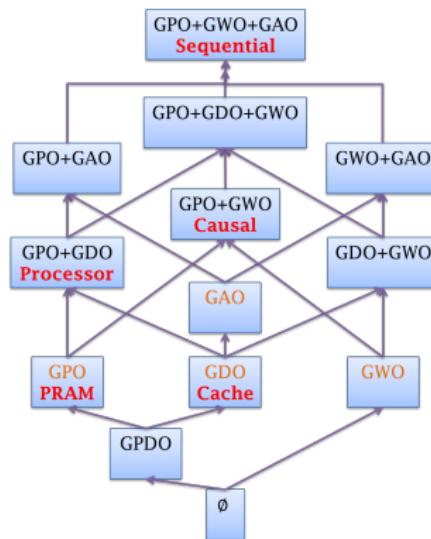
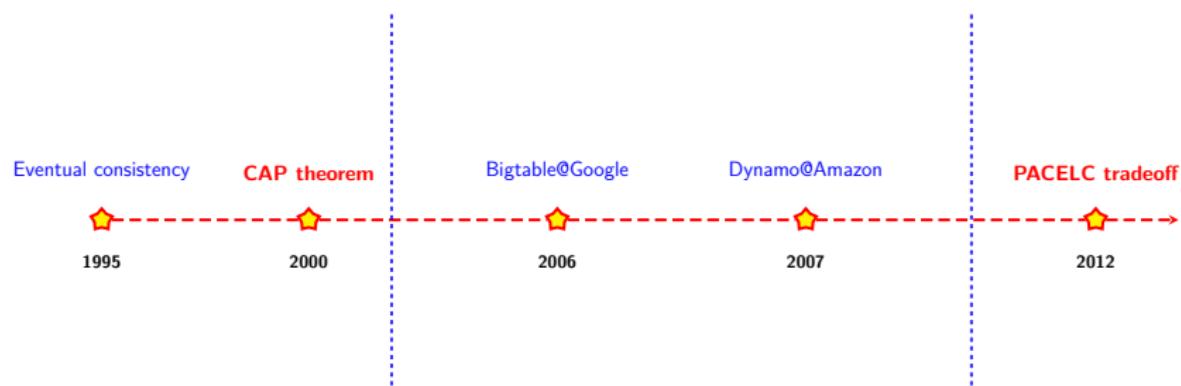


图: 一致性模型. 依据 [Steinke@JACM'04] 中 Fig.13 重绘.

核心 tradeoff: 一致性模型的计算能力 vs. 系统性能

数据一致性问题研究的历史阶段 (分布式系统)



数据一致性问题研究的历史阶段 (分布式系统)



Eventual consistency



1995

CAP theorem



2000

Bigtable@Google



2006

Dynamo@Amazon



2007

PACELC tradeoff



2012

Managing Update Conflicts in Bayou,
a Weakly Connected Replicated Storage System

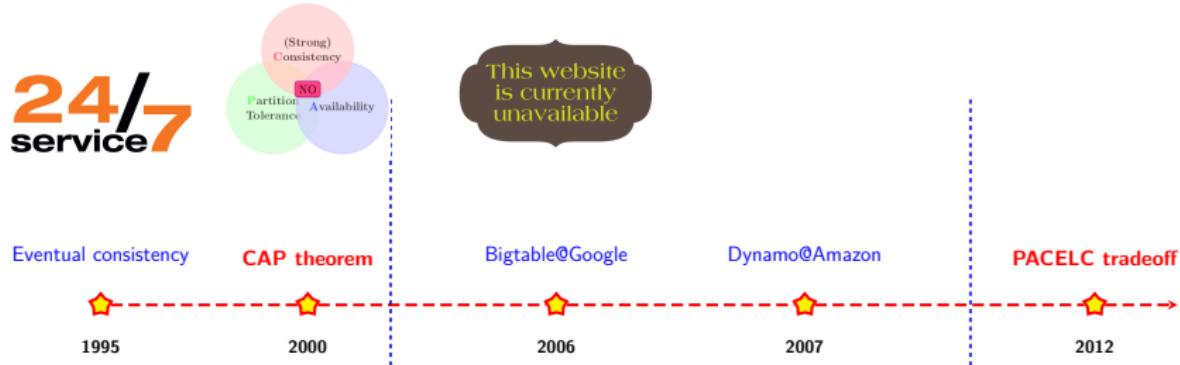
Douglas B. Terry, Marvin M. Thurber, Kara Barnes, Alan J. Demers,
Mike J. Spreitzer and Carl H. Hauser

Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304 U.S.A.

Towards Robust
Distributed Systems

Dr. Eric A. Brewer
Professor, UC Berkeley
Co-Founder & Chief Scientist, Inktomi

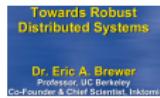
数据一致性问题研究的历史阶段 (分布式系统)



Managing Update Conflicts in Bayou,
a Weakly Connected Replicated Storage System

Douglas B. Terry, Marvin M. Thurber, Kara Barneser, Alan J. Demers,
Mike J. Spreitzer and Carl H. Hauser

Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304 U.S.A.

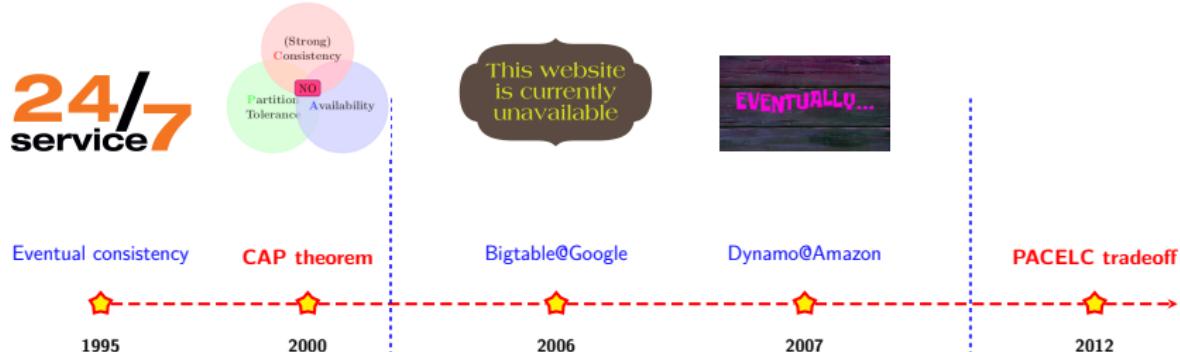


Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
[jeff.dean,sanjay.ghemawat,wilson.hsieh,deborah.wallach,mike.burrows,tushar.chandra,andy.fikes,robert.gruber]@google.com

Google, Inc.

数据一致性问题研究的历史阶段 (分布式系统)



Managing Update Conflicts in Bayou,
a Weakly Connected Replicated Storage System
Douglas B. Terry, Marvin M. Thurber, Kara Barnes, Alan J. Demers,
Mike J. Spreitzer and Carl H. Hauser

Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304 U.S.A.

Towards Robust
Distributed Systems
Dr. Eric A. Brewer
Professor, UC Berkeley
Co-Founder & Chief Scientist, Inkster

Bigtable: A Distributed Storage System for Structured Data
Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Heid, David A. Hirschberg, Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Ketukapat, Arvind Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
[bigtable-sigmod06.pdf]@csail.mit.edu, gpc@google.com
Google, Inc.

Dynamo: Amazon's Highly Available Key-value Store
Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Ketukapat, Arvind Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
[dynamo-sosp07.pdf]@amazon.com

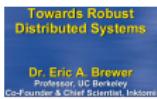
数据一致性问题研究的历史阶段 (分布式系统)



Managing Update Conflicts in Bayou,
a Weakly Connected Replicated Storage System

Douglas B. Terry, Marvin M. Thurber, Kara Barneser, Alan J. Demers,
Mike J. Spreitzer and Carl H. Hauser

Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304 U.S.A.



Bigtable: A Distributed Storage System for Structured Data

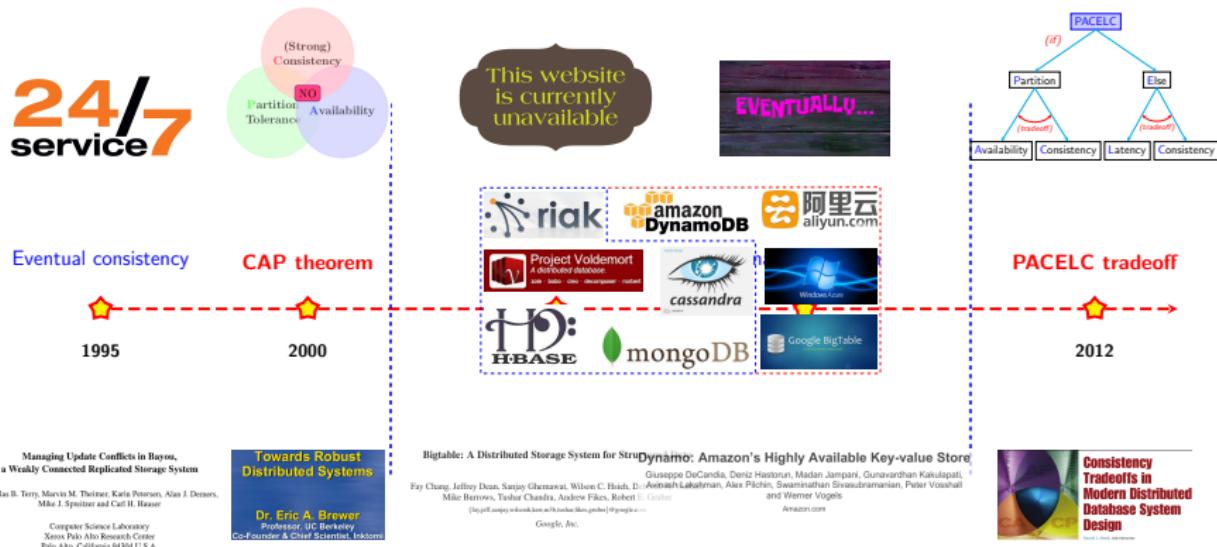
Dynamo: Amazon's Highly Available Key-value Store

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Heid, D. Aminish Lakshman, Alex Pitkänen, Swaminathan Sivasubramanian, Peter Vosshall, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, Giuseppe DeCandia, Dorin Hastorun, Madan Jampani, Gunavardhan Katulapati, and Werner Vogels

{fay,jeff,deej,sanjay,wilson,alex,swami,peter,mike,tushar,andy,robert}@google.com

Google, Inc.

数据一致性问题研究的历史阶段 (分布式系统)



数据一致性问题研究的历史阶段 (结论)

新平台的两个特点：

需要什么样的数据一致性理论？

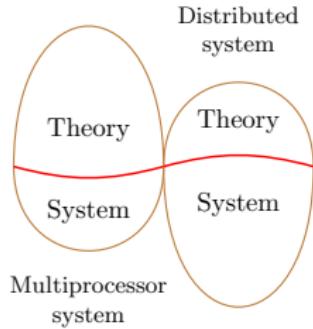
数据一致性问题研究的历史阶段 (结论)

新平台的两个特点:

- (1) 云计算新平台凸显应用价值观

需要什么样的数据一致性理论?

- (1) 与应用价值观相匹配



数据一致性问题研究的历史阶段 (结论)

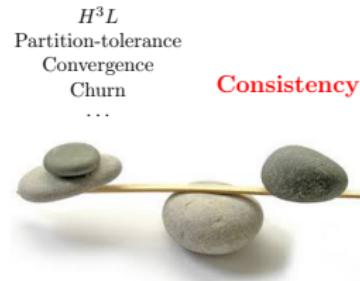
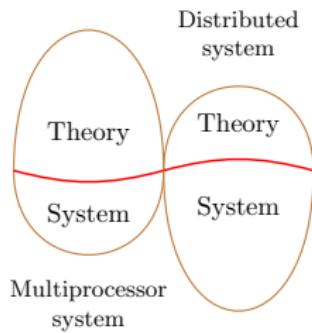
新平台的两个特点:

- (1) 云计算新平台凸显应用价值观
- (2) 应用价值观积极拥抱 tradeoffs

需要什么样的数据一致性理论?

(1) 与应用价值观相匹配

(2) 体现更丰富的 tradeoffs



数据一致性问题研究的发展趋势及我们的工作 (I)

购物车一致性需求

- ▶ 优先 `read-my-writes`
- ▶ 可接受 `any consistency`
只要延迟低于 300ms

出租车实时位置查询一致性需求:

- ▶ 所有读请求都要满足 `2-atomicity`
- ▶ 违反 `atomicity` 的读请求低于 1%

数据一致性问题研究的发展趋势及我们的工作 (I)

购物车一致性需求

- ▶ 优先 `read-my-writes`
- ▶ 可接受 `any consistency`
只要延迟低于 300ms

出租车实时位置查询一致性需求:

- ▶ 所有读请求都要满足 `2-atomicity`
- ▶ 违反 `atomicity` 的读请求低于 1%

应用价值观导向的数据一致性理论:

1. 多样化, 可调节
2. 精细化, 可度量

数据一致性问题研究的发展趋势及我们的工作 (II)

多样化: 从单一到融合 (mono- vs. multi-) [Terry@CACM'13]

- ▶ 融合强弱一致性: 不同操作, 不同一致性需求
- ▶ 融合一致与不一致: 容忍“有限度”的不一致



数据一致性问题研究的发展趋势及我们的工作 (II)

多样化: 从单一到融合 (mono- vs. multi-) [Terry@CACM'13]

- ▶ 融合强弱一致性: 不同操作, 不同一致性需求
- ▶ 融合一致与不一致: 容忍“有限度”的不一致

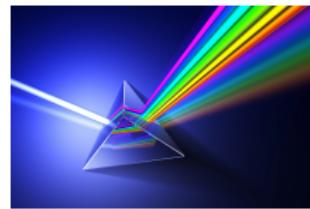


可调节: think *dynamically* [Terry@SOSP'13]

依据应用需求/系统状态调节数据一致性

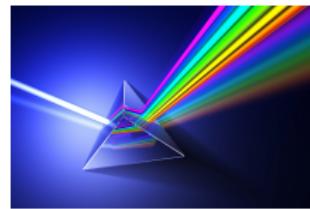
数据一致性问题研究的发展趋势及我们的工作 (III)

精细化: 从二元到连续谱 [Yu@TOCS'02]



数据一致性问题研究的发展趋势及我们的工作 (III)

精细化: 从二元到连续谱 [Yu@TOCS'02]



可度量: think *probabilistically* [Brewer@PODC'00]



量化系统执行, 后验系统对一致性的满足程度

数据一致性问题研究的发展趋势及我们的工作 (III)

2 个理念:

1. 多样化, 可调节
2. 精细化, 可度量

3 份工作:

1. VPC
2. PA2AM
3. RVSI

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

- 理论模型: 分布共享数据
- 技术框架

4 主要工作

5 未来工作

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

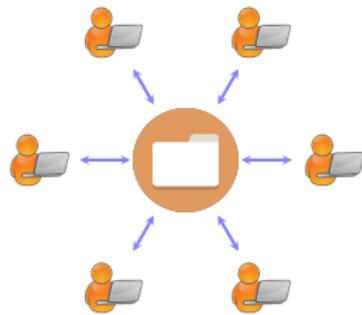
3 研究方法

- 理论模型: 分布共享数据
- 技术框架

4 主要工作

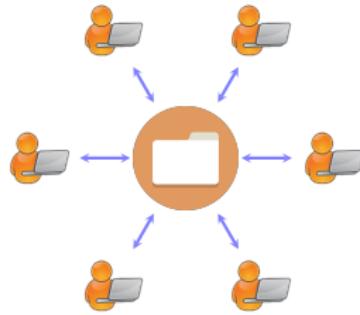
5 未来工作

分布共享数据服务

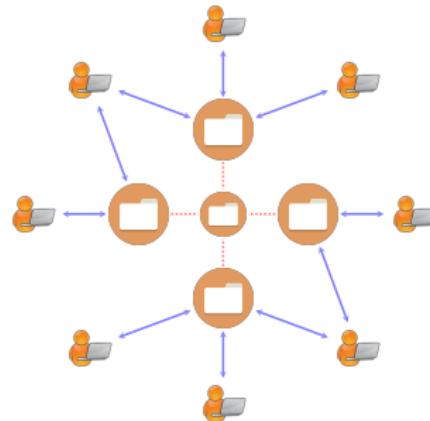


共享数据系统 (single copy)

分布共享数据服务

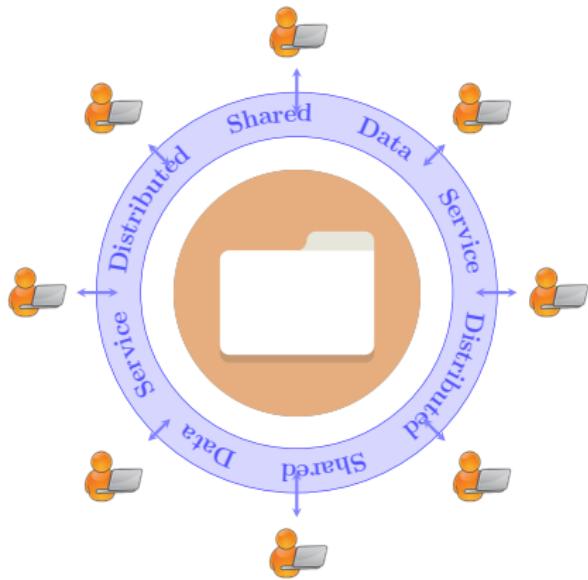


共享数据系统 (single copy)



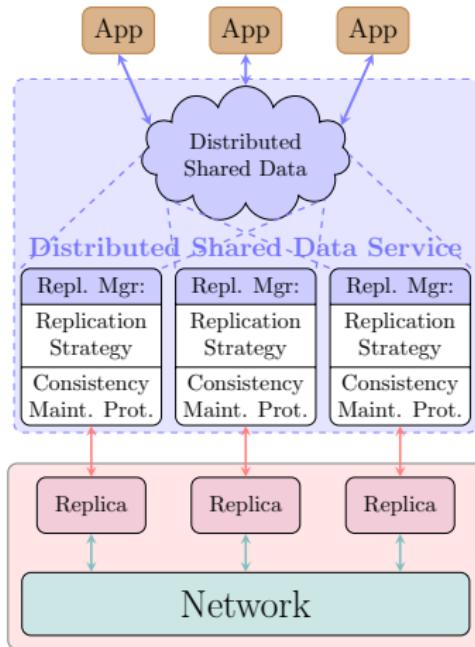
分布数据系统 (replicas)

分布共享数据服务



分布共享数据服务作为中间件管理分布数据

分布共享数据服务



分布共享数据服务 (中间件): 在分布数据之上提供共享数据的抽象

分布共享数据服务 (注)

分布共享内存模型 (多处理器系统)

[传统概念]

+

分布数据系统

[新平台]

MORE OLD WINE
in
NEW BOTTLES



Gordon Jacob
1895-1984

2 flutes, 2 oboes, 2 clarinets, 2 bassoons
contrabassoon, 2 horns, 2 trumpets

Emerson Edition
93

分布共享数据服务 (注)

分布共享内存模型 (多处理器系统)

[传统概念]

+

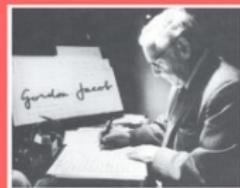
分布数据系统

[新平台]

新平台凸显应用价值观:

1. 多样化, 可调节
2. 精细化, 可度量

MORE OLD WINE
in
NEW BOTTLES



Gordon Jacob
1895-1984

2 flutes, 2 oboes, 2 clarinets, 2 bassoons
contrabassoon, 2 horns, 2 trumpets

Emerson Edition
93

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

- 理论模型: 分布共享数据
- 技术框架

4 主要工作

5 未来工作

分布数据一致性问题

分布数据一致性问题：

- ✓ 分布：分区 + 副本
- ✗ 数据：数据类型
- ✗ 一致性：关键问题

分布数据一致性问题

分布数据一致性问题:

- ✓ 分布: 分区 + 副本
- ✗ 数据: 数据类型
- ✗ 一致性: 关键问题

数据类型:

- ▶ 单独的变量 (x, y)
- ▶ 数据结构 (SET, LIST)
- ▶ 事务 (Tx)

分布数据一致性问题

分布数据一致性问题:

- ✓ 分布: 分区 + 副本
- ✗ 数据: 数据类型
- ✗ 一致性: 关键问题

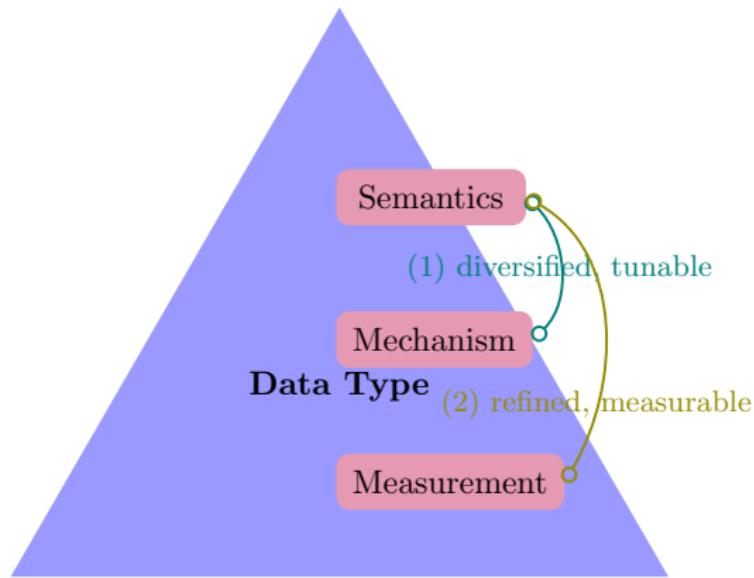
数据类型:

- ▶ 单独的变量 (x, y)
- ▶ 数据结构 (SET, LIST)
- ▶ 事务 (Tx)

一致性关键问题:

- ▶ 模型 (semantics; 是什么)
- ▶ 机制 (mechanism; 怎么做)
- ▶ 度量 (measurement; 怎么样)

技术框架



数据类型

数据类型：从个体到群组

数据类型

数据类型：从个体到群组

- ▶ 单独读写变量

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

数据类型

数据类型：从个体到群组

- ▶ 单独读写变量
- ▶ 事务对象
 - ▶ 事务 \triangleq 多个读写变量的操作序列
 - ▶ 支持“all-or-none”写语义
 - ▶ 易于开发并发应用

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623



一致性模型

一致性模型 [Steinke@JACM'04] [Adya@Thesis'99]:

- ▶ 多进程并发操作某数据类型
- ▶ 规定各操作的语义
 - ▶ 读写变量: 读操作允许的返回值
 - ▶ 事务对象: 事务创建与提交操作的语义

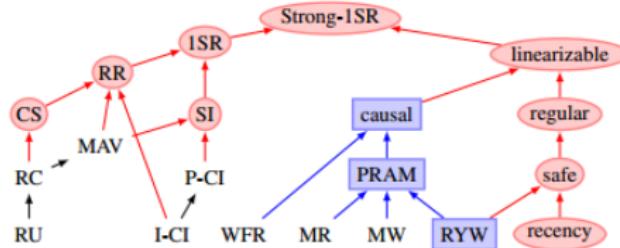


图: 来自 [Bailis@VLDB'14]

一致性模型

一致性模型的集合定义：

一致性模型

一致性模型的集合定义：

系统执行 $e \triangleq$ 该执行所产生的事件的序列



一致性模型

一致性模型的集合定义：

系统执行 $e \triangleq$ 该执行所产生的事件的序列

分布式系统 $S \triangleq \{\text{该系统的所有可能执行}\}$



一致性模型

一致性模型的集合定义：

系统执行 $e \triangleq$ 该执行所产生的事件的序列

分布式系统 $S \triangleq \{\text{该系统的所有可能执行}\}$

一致性模型 $C \triangleq \{\text{该模型所允许的所有系统执行}\}$



一致性实现机制

给定一致性模型 \mathcal{C} , 设计系统 \mathcal{S} :

$$\forall e \in \mathcal{S} : e \in \mathcal{C}.$$

i.e., $\mathcal{S} \subseteq \mathcal{C}$.



一致性实现机制

给定一致性模型 \mathcal{C} , 设计系统 \mathcal{S} :

$$\forall e \in \mathcal{S} : e \in \mathcal{C}.$$

i.e., $\mathcal{S} \subseteq \mathcal{C}$.



SPECS

“多样化, 可调节”的难点:

- ▶ 兼容的混合一致性模型
- ▶ 实现手段之一: 参数化



一致性度量方法

给定系统 S 及一致性模型 \mathcal{C} ,

一致性度量方法

给定系统 \mathcal{S} 及一致性模型 \mathcal{C} ,

对于 $e \in \mathcal{S}$:

验证 (verify): $e \in \mathcal{C}?$ $\Rightarrow \{0, 1\}$

量化 (quantify): $e \in \mathcal{C}?$ $\Rightarrow (0, 1)$



一致性度量方法

给定系统 \mathcal{S} 及一致性模型 \mathcal{C} ,

对于 $e \in \mathcal{S}$:

验证 (verify): $e \in \mathcal{C}?$ $\Rightarrow \{0, 1\}$

量化 (quantify): $e \in \mathcal{C}?$ $\Rightarrow (0, 1)$



“精细化, 可度量”的难点:

验证: 算法设计

量化: 数学建模

"All models are wrong, but some are useful."
- George Box

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

- 概述
- VPC: Pipelined-RAM 一致性验证
- PA2AM: Atomicity 一致性维护与量化
- RVSI: Snapshot Isolation 一致性弱化与维护

5 未来工作

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

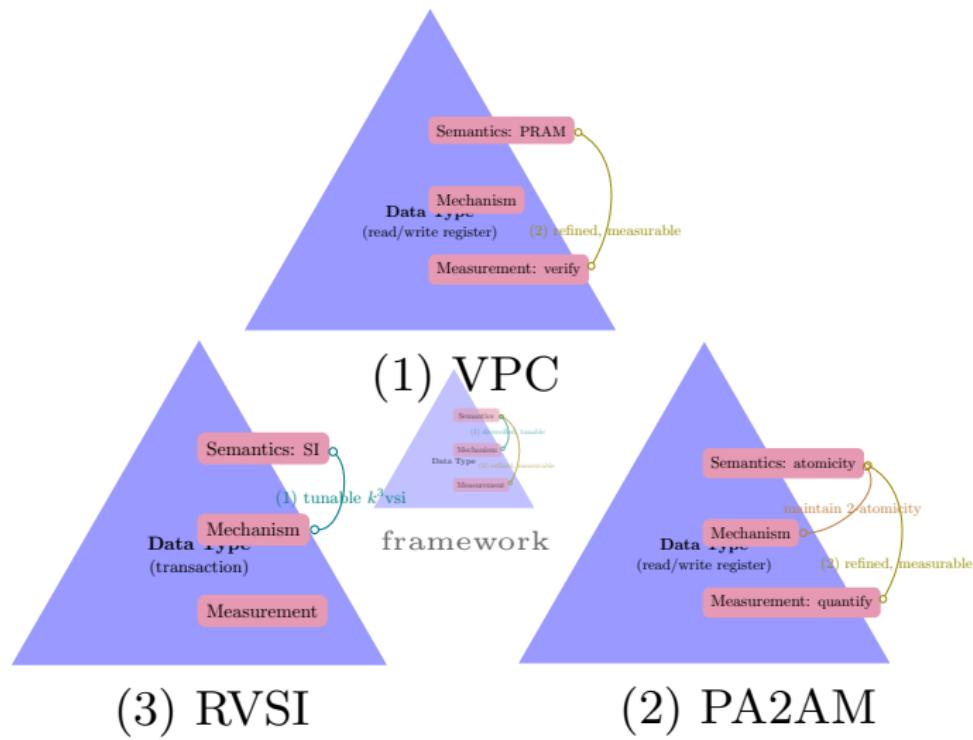
4 主要工作

• 概述

- VPC: Pipelined-RAM 一致性验证
- PA2AM: Atomicity 一致性维护与量化
- RVSI: Snapshot Isolation 一致性弱化与维护

5 未来工作

工作概述



分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

- 概述

- VPC: Pipelined-RAM 一致性验证
- PA2AM: Atomicity 一致性维护与量化
- RVSI: Snapshot Isolation 一致性弱化与维护

5 未来工作

在研究框架中的位置

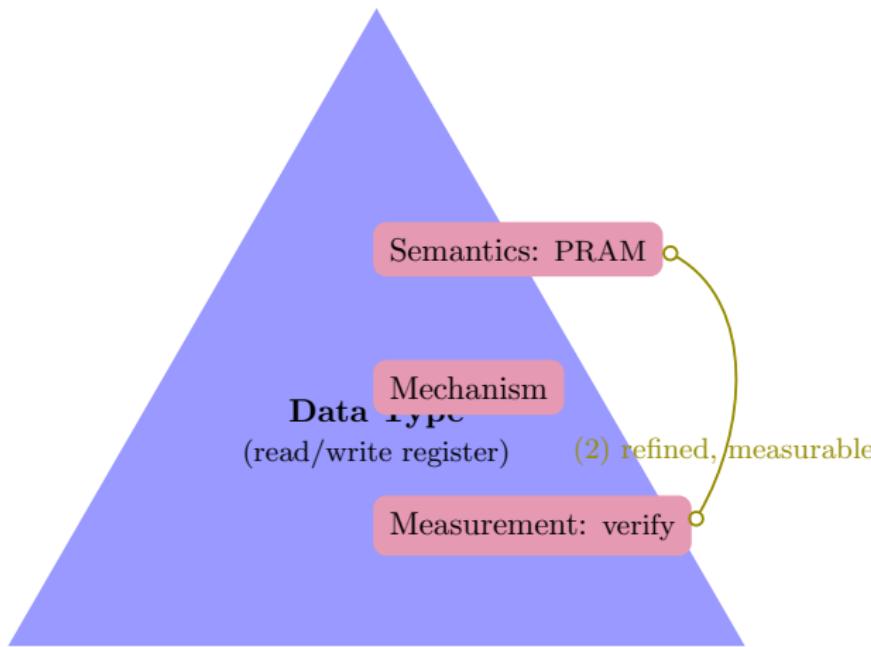


图: VPC — Pipelined-RAM 一致性验证.

研究动机

问题: 为什么要验证 Pipelined-RAM (PRAM) 一致性?

验证: 用户需确认存储系统提供了其所声称的数据一致性

[Golab@PODC'11] [Facebook@SOSP'15]

- ▶ 商用存储系统对于用户是黑盒
- ▶ 后验分析系统执行
- ▶ SLA 服务补偿 [Amazon@SOSP'07]

PRAM: 存储系统常提供“会话”(session) 一致性 [Saito@CSUR'05]

[Terry@CACM'13]

- ▶ 包含了弱一致性的诸多变体
- ▶ 近似于 PRAM 一致性 [Bailis@VLDB'13]

VPC 问题定义

定义 (VPC: Verifying PRAM Consistency)

VPC 判定问题:

- 实例: ▶ 系统执行 (*execution e*; 即, 读写操作序列)
▶ *PRAM* 一致性模型 (\mathcal{C})

- 问题: ▶ 该执行是否满足 *PRAM* 一致性模型
($e \in \mathcal{C} \Rightarrow \{0, 1\}$)?

对 VPC 问题的系统性研究

	<i>(S)ingle variable</i>	<i>(M)ultiple variables</i>
<i>write (D)uplicate values</i>	VPC-SD (NPC) [*]	VPC-MD (NPC) [*]
<i>write (U)nique value</i>	VPC-SU (P) [Golab@PODC'11]	VPC-MU (P) [*]

表: VPC 问题的四种变体 (按“执行”的类型) 及验证复杂性结果
[*] : new results).

分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

- 概述
- VPC: Pipelined-RAM 一致性验证
- PA2AM: Atomicity 一致性维护与量化
- RVSI: Snapshot Isolation 一致性弱化与维护

5 未来工作

在研究框架中的位置

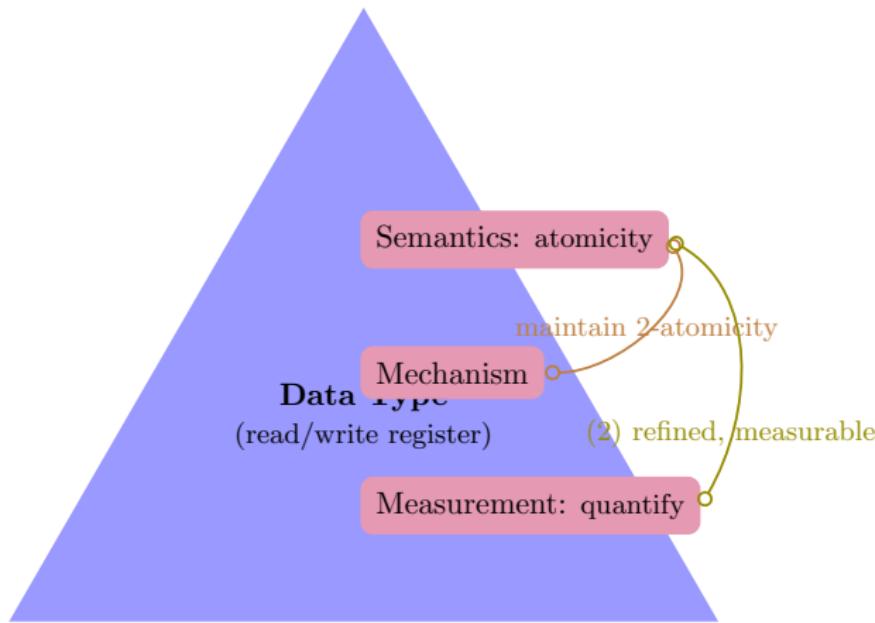


图: 2AM — Atomicity 一致性维护与量化.

研究动机

问题: 为什么要提出 probabilistically-atomic 2-atomicity 一致性?

“数据一致性/访问延迟” PACELC 权衡 [Abadi@IEEE Computer'12]:



为保证低延迟, 采用较弱一致性:

“100ms of additional latency = 1% drop in sales” – [Amazon'06]

系统	一致性
Dynamo@Amazon	eventual consistency
Tao@Facebook	read-after-write
PNUTS@Yahoo!	cache consistency

2-atomicity 一致性

2AM: 在保证低延迟的情况下获得尽可能强的数据一致性.

定义 (2-atomicity 一致性)

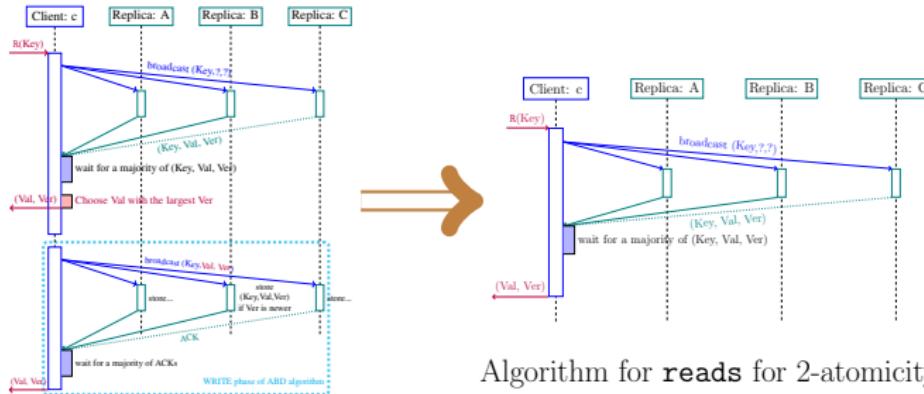
低延迟: 读操作只需一轮网络通信

尽可能强: 对 *atomicity* (最强) 的弱化

- ▶ (版本) 允许读陈旧值, 且陈旧度 $k \leq 2$
- ▶ (概率) $\mathbb{P}(k = 2)$ 很小

2AM 维护算法

2AM (单写多读) 维护算法: 读 (写) 只需一轮网络通信



Algorithm for **reads** for atomicity.

Algorithm for **reads** for 2-atomicity.

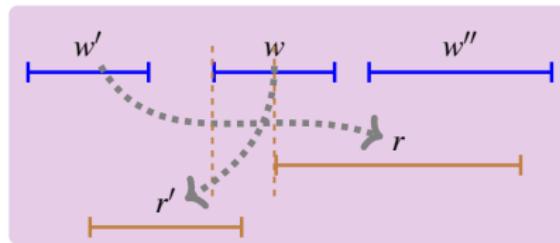
图: 经典 atomicity 算法中, 读操作需两轮网络通信 [ABD@JACM'95]

[Dutta@PODC'04]. 2AM 算法中, 读操作只需一轮网络通信: 读取半数以上副本节点, 返回最新值.

2AM 量化分析

问题: 2AM 算法在多大程度上违反了 atomicity?

- ▶ 充要条件: ONI (old-new inversion)



- ▶ 2AM 量化分析: 计算 $\mathbb{P}(\text{ONI})$, 其值越小越好
 1. $\text{ONI} \triangleq \text{CP} \cap \text{RWP}$
 2. 排队论建模, 计算 $\mathbb{P}(\text{CP})$
 3. 带时间的球盒模型, 计算 $\mathbb{P}(\text{RWP}|\text{CP})$

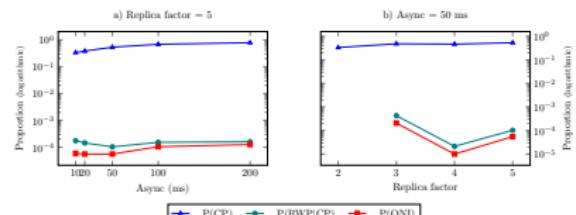
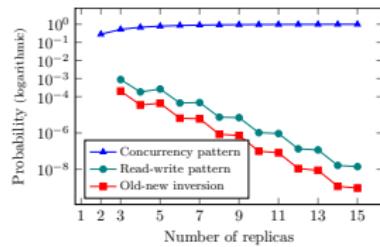
2AM 量化分析

公式推导:

$$\begin{aligned}\mathbb{P}\{\text{CP} \mid R' = m\} &= \mathbb{P}(E_{N-1,m}) \\ &= \sum_{k=0}^{N-2} \binom{N-1}{k} \binom{m-1}{N-k-2} p_0^k r^{N-k-1} s^m\end{aligned}$$

$$\begin{aligned}\mathbb{P}\{\text{RWP} \mid R' = m\} &\leq \mathbb{P}\{r \neq R(w)\} \times \left(1 - \mathbb{P}\{r' \neq R(w) \mid r \neq R(w)\}^m\right) \\ &\leq e^{-q\lambda_w t} \frac{\alpha^q B(q, \alpha(n-q)+1)}{B(q, n-q+1)} \\ &\quad \cdot \left(1 - \left(\frac{J_1}{B(q, n-q+1)}\right)^m\right).\end{aligned}\quad (4)$$

数值结果 (左一) 与实验结果 (右二):



分布数据一致性理论与技术研究

1 研究背景

2 研究问题

3 研究方法

4 主要工作

- 概述
- VPC: Pipelined-RAM 一致性验证
- PA2AM: Atomicity 一致性维护与量化
- RVSI: Snapshot Isolation 一致性弱化与维护

5 未来工作

在研究框架中的位置

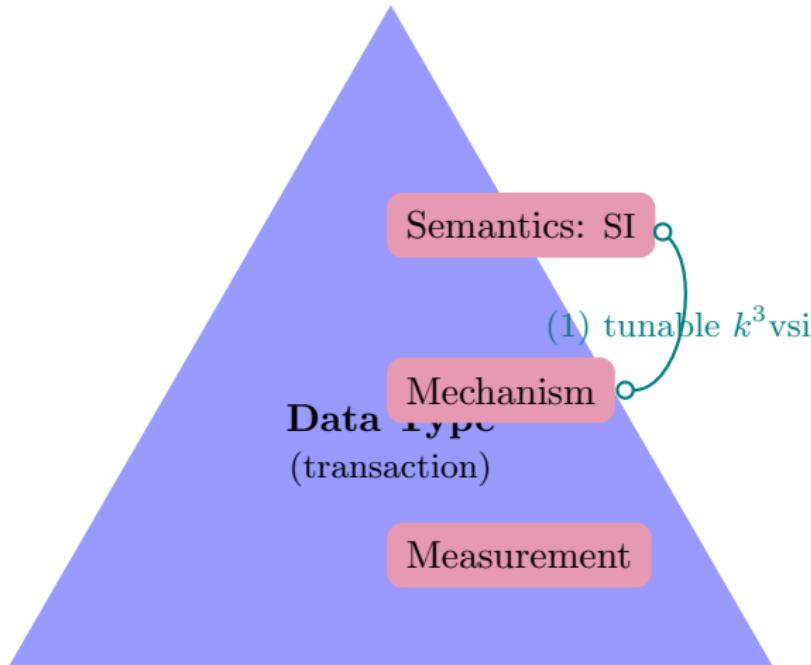


图: RVSI — Snapshot Isolation 一致性弱化与维护

研究动机

问题: 为什么要提出 RVI 一致性?

分布式事务:

- ▶ “all-or-none” 语义
- ▶ 受到分布式存储系统的关注
[Cassandra@CASSANDRA-ISSUE-7056'14]

弱一致性:

- ▶ PCSI [Elnekety@SRDS'05] SI [Lin@TODS'09]
PSI [Sovran@SOSP'11] NMSI [Ardekani@SRDS'13]

研究动机

问题: 为什么要提出 RVI 一致性?

- 分布式事务:**
 - ▶ “all-or-none” 语义
 - ▶ 受到分布式存储系统的关注
[Cassandra@CASSANDRA-ISSUE-7056'14]
- 弱一致性:**
 - ▶ PCSI [Elnekety@SRDS'05] SI [Lin@TODS'09]
 - ▶ PSI [Sovran@SOSP'11] NMSI [Ardekani@SRDS'13]
- 异常控制:**
 - ▶ 容忍“有限度的”异常 [Yu@TOCS'02]
- 可定制:**
 - ▶ 不同应用对一致性需求不同 [Terry@CACM'13]
 - ▶ 运行时决定 [Terry@SOSP&TR'13]

研究动机

问题: 为什么要提出 RVSI 一致性?

- 分布式事务:**
 - ▶ “all-or-none” 语义
 - ▶ 受到分布式存储系统的关注
[Cassandra@CASSANDRA-ISSUE-7056'14]
- 弱一致性:**
 - ▶ PCSI [Elnekety@SRDS'05] SI [Lin@TODS'09]
 - ▶ PSI [Sovran@SOSP'11] NMSI [Ardekani@SRDS'13]
- 异常控制:**
 - ▶ 容忍“有限度的”异常 [Yu@TOCS'02]
- 可定制:**
 - ▶ 不同应用对一致性需求不同 [Terry@CACM'13]
 - ▶ 运行时决定 [Terry@SOSP&TR'13]

RVSI (Relaxed Version Snapshot Isolation):

1. 支持可定制一致性
2. 提供“有限度的”异常控制
3. 支持高效的分布式实现

RVSI 定义

RVSI 定义原则:

- ▶ 参数 k_1, k_2, k_3 控制“异常”程度
- ▶ $\text{RC} \supset \text{RVSI}(k_1, k_2, k_3) \supset \text{SI}$
- ▶ $\text{RVSI}(\infty, \infty, \infty) = \text{RC}; \quad \text{RVSI}(1, 0, *) = \text{SI}$

定义 (RVSI: Relaxed Version Snapshot Isolation)

单变量读 $\text{read}(x)$:

1. 允许读 $\leq k_1$ 陈旧值
2. 允许读 $\leq k_2$ 并发更新

多变量读 $\text{read}(x), \text{read}(y)$:

3. $\text{dist}(x, y) \leq k_3$

RVSI 维护算法

$$\text{RC} \supset \text{RVSI}(k_1, k_2, k_3) \supset \text{SI}$$

RVSI 维护算法:

- ▶ 以分布式 RC 和 SI 协议为基础
- ▶ 事务执行时, 添加 RVSI “版本约束” (k_1, k_2, k_3 相关)
- ▶ 事务提交时, 检查 RVSI “版本约束”

RVSI 维护算法

$$\text{RC} \supset \text{RVSI}(k_1, k_2, k_3) \supset \text{SI}$$

RVSI 维护算法:

- ▶ 以分布式 RC 和 SI 协议为基础
- ▶ 事务执行时, 添加 RVSI “版本约束” (k_1, k_2, k_3 相关)
- ▶ 事务提交时, 检查 RVSI “版本约束”

RVSI 实验:

- ▶ Chameleon: a distributed, partitioned, replicated, transactional key-value store
- ▶ 阿里云部署

<https://github.com/hengxin/chameleon-transactional-kvstore>

分布数据一致性理论与技术研究

1 研究背景

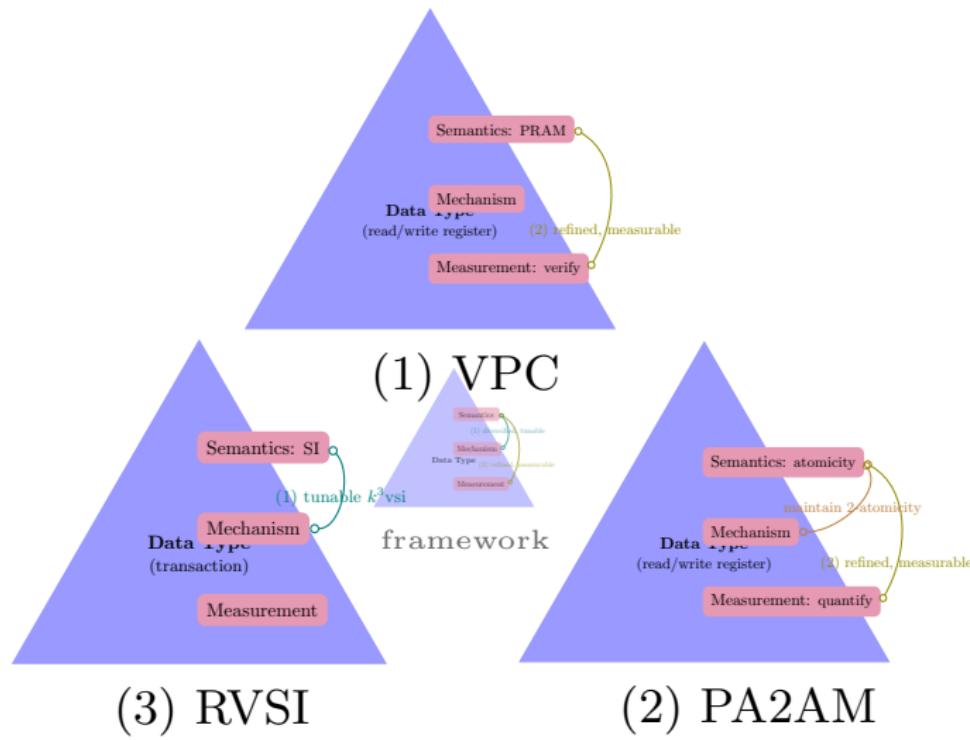
2 研究问题

3 研究方法

4 主要工作

5 未来工作

工作总结



未来工作

多样化, 可定制; 精细化, 可度量:

1. 支持“多写”的 atomic 变量 (扩展 2AM 工作)
2. 支持 P2P 架构的“一致性可定制”实现
3. 事务与非事务一致性模型的统一框架

未来工作

多样化, 可定制; 精细化, 可度量:

1. 支持“多写”的 atomic 变量 (扩展 2AM 工作)
 - ▶ **前提:** 读操作只需一轮网络通信 (*fast read*)
 - ▶ **理论问题:** 是否存在允许 fast read 的 (k -)atomicity 算法?
 - ▶ **可度量:** 如何定义 & 量化 p AM (p : probabilistic)?
2. 支持 P2P 架构的“一致性可定制”实现
3. 事务与非事务一致性模型的统一框架

未来工作

多样化, 可定制; 精细化, 可度量:

1. 支持“多写”的 atomic 变量 (扩展 2AM 工作)
2. 支持 P2P 架构的“一致性可定制”实现
 - ▶ **已有工作:** 非事务, 可定制, master-slave 架构 [Terry@SOSP'13]
 - ▶ **动机:** Cassandra 采用 P2P 架构 [Facebook@SIGOPS OSR'10]
 - ▶ **可定制:** 一致性模型的兼容性与重定义
3. 事务与非事务一致性模型的统一框架

未来工作

多样化, 可定制; 精细化, 可度量:

1. 支持“多写”的 atomic 变量 (扩展 2AM 工作)

2. 支持 P2P 架构的“一致性可定制”实现

3. 事务与非事务一致性模型的统一框架

- ▶ **异:** “all-or-none”语义
- ▶ **同:** 操作间序关系
- ▶ **多样化:** 更丰富, 更结构化的一致性模型

未来工作

多样化, 可定制; 精细化, 可度量:

1. 支持“多写”的 atomic 变量 (扩展 2AM 工作)

- ▶ **前提:** 读操作只需一轮网络通信 (*fast read*)
- ▶ **理论问题:** 是否存在允许 fast read 的 (k -)atomicity 算法?
- ▶ **可度量:** 如何定义 & 量化 p AM (p : probabilistic)?

2. 支持 P2P 架构的“一致性可定制”实现

- ▶ **已有工作:** 非事务, 可定制, master-slave 架构 [Terry@SOSP'13]
- ▶ **动机:** Cassandra 采用 P2P 架构 [Facebook@SIGOPS OSR'10]
- ▶ **可定制:** 一致性模型的兼容性与重定义

3. 事务与非事务一致性模型的统一框架

- ▶ **异:** “all-or-none”语义
- ▶ **同:** 操作间序关系
- ▶ **多样化:** 更丰富, 更结构化的一致性模型



hengxin0912@gmail.com



未来工作

数据一致性问题的发展趋势:

1. give more consideration to SLA ⇒ 应用价值观导向的数据一致性
(我们追随的发展趋势)
2. poor definition of consistency ⇒ 为弱一致性奠定理论基础
3. poor understanding of boundaries ⇒ 探索更强的数据一致性模型及理论界限