

分布数据一致性技术研究

魏恒峰

导师: 吕建 黄宇

南京大学软件所

November 5, 2016

分布数据一致性技术研究

① 研究背景

② 研究问题

③ 相关工作

④ 技术框架

分布数据一致性技术研究

① 研究背景

② 研究问题

③ 相关工作

④ 技术框架

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

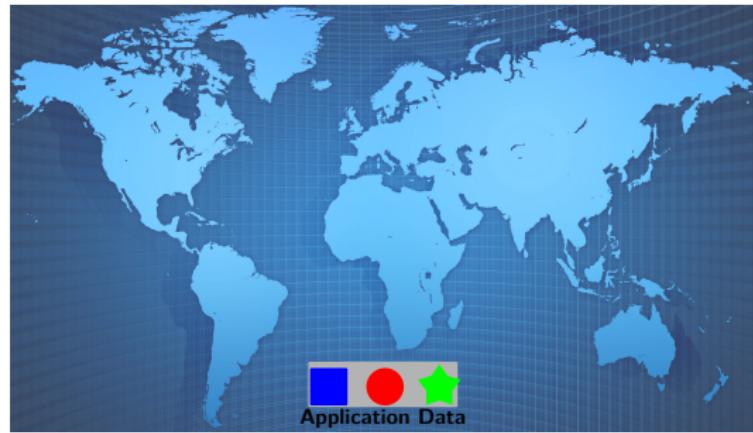
底层数据服务系统特性需求 (H^3L):

- ▶ 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.

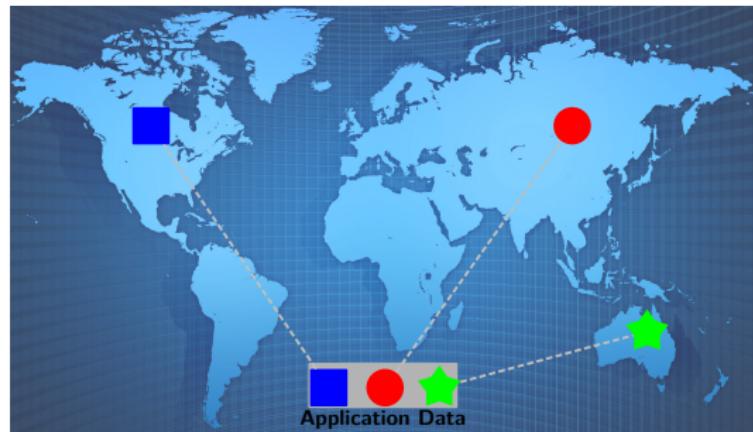
² 数据来自 InfoQ.

分布数据



应用数据:

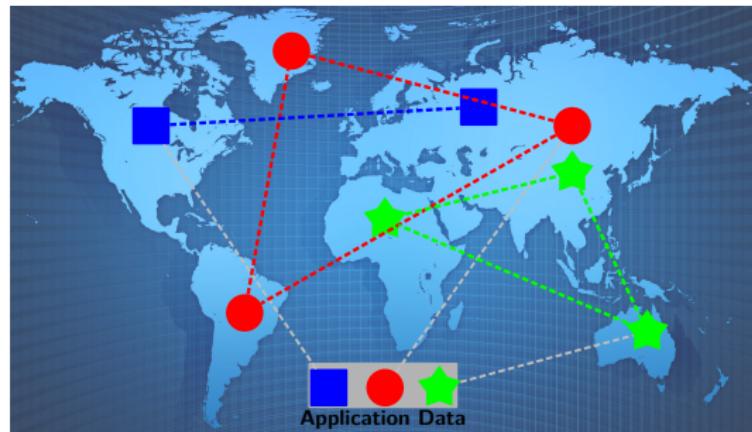
分布数据



应用数据:

1. 分区 (partition): 水平扩展

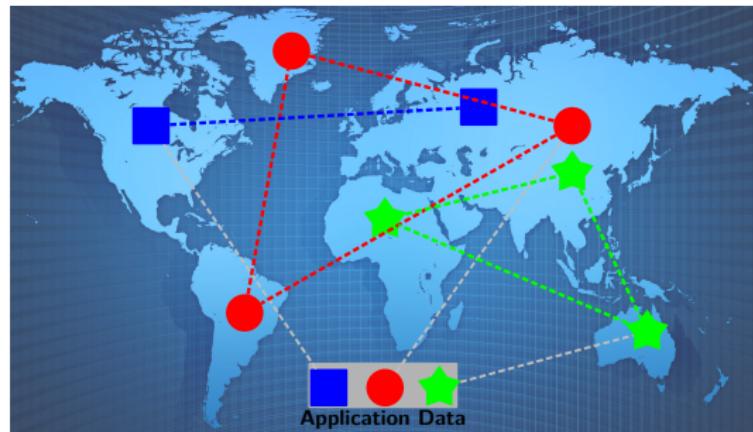
分布数据



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

分布数据一致性技术研究

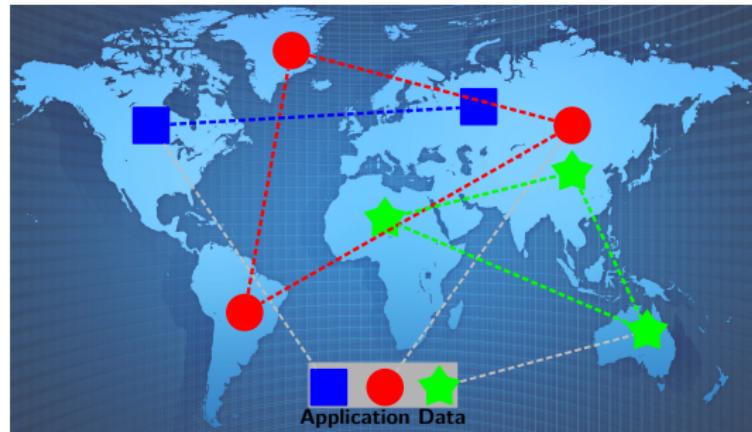
1 研究背景

2 研究问题

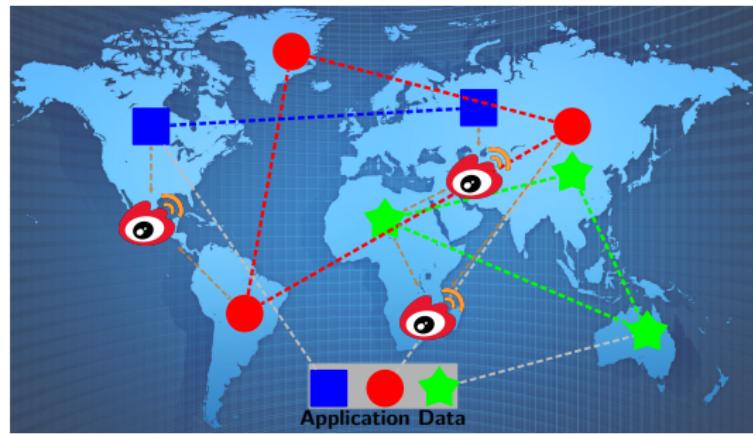
3 相关工作

4 技术框架

分布数据一致性问题

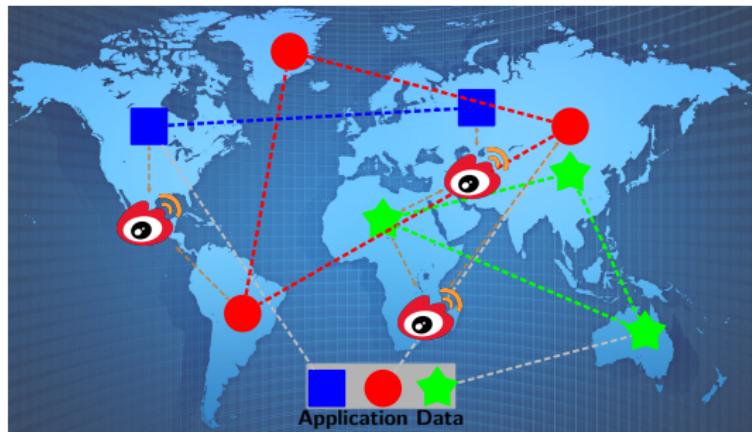


分布数据一致性问题



分布数据 (distributed data) \Leftarrow 共享 (集中式) 数据 (shared data):

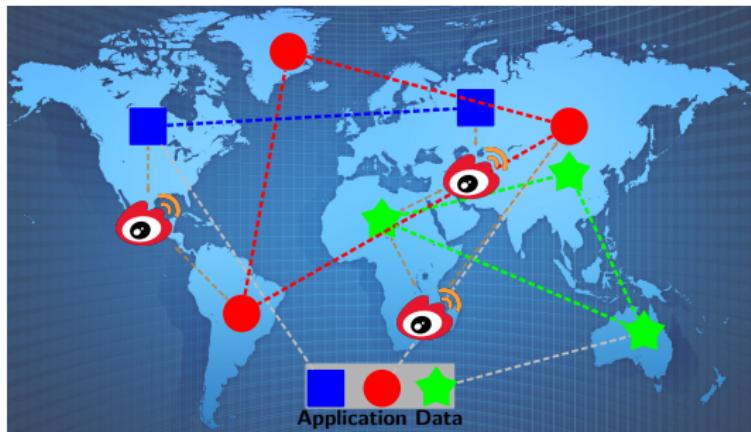
分布数据一致性问题



分布数据 (distributed data) \Leftarrow 共享 (集中式) 数据 (shared data):

上层应用: 分布数据的语义是什么? 如何“方便”地使用分布数据?

分布数据一致性问题



分布数据 (distributed data) \Leftarrow 共享 (集中式) 数据 (shared data):

上层应用: 分布数据的语义是什么? 如何“方便”地使用分布数据?

数据副本: 以何种顺序应用更新? 对应用提供什么保证?

分布数据一致性问题

Replica A Replica B

图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

分布数据一致性问题

Alice: I've **lost** my ring.

Alice: I **found** it upstairs.

Bob: **Glad** to hear that.

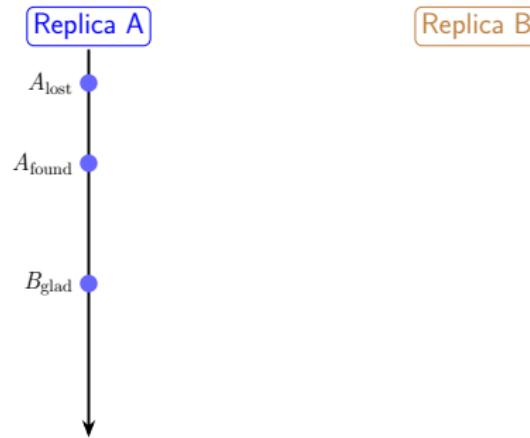


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

分布数据一致性问题

Alice: I've **lost** my ring.

Alice: I **found** it upstairs.

Bob: **Glad** to hear that.

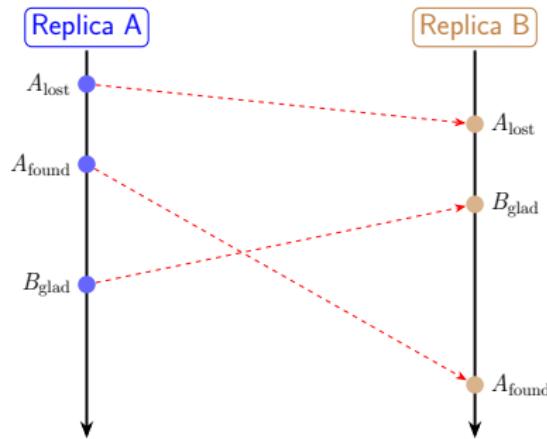


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

分布数据一致性问题

Alice: I've **lost** my ring.

Alice: I **found** it upstairs.

Bob: **Glad** to hear that.

Alice: I've **lost** my ring.

Bob: **Glad** to hear that.

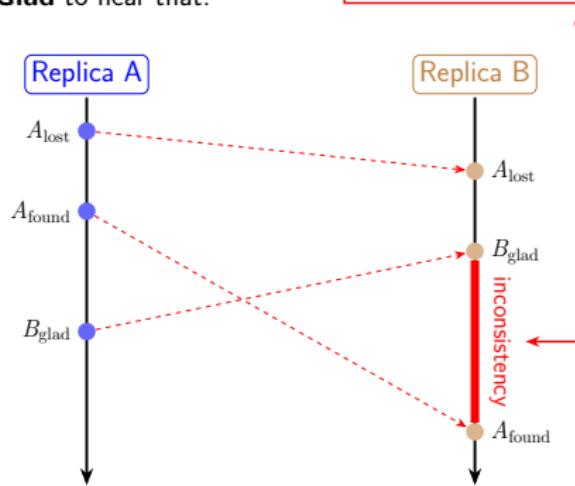


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

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) [Guerraoui@TCDE'16]:

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) [Guerraoui@TCDE'16]:

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

Consistency



以数据一致性为核心的权衡使得该问题具有挑战性

分布数据一致性问题

论文研究问题：

考虑到上述权衡，
面向大规模分布式系统的
分布数据一致性理论应具有什么特性？

分布数据一致性问题

论文研究问题:

考虑到上述权衡，
面向大规模分布式系统的
分布数据一致性理论应具有什么特性？

考察分布数据一致性问题研究的历史：

- ▶ 核心权衡与解决方案

分布数据一致性问题

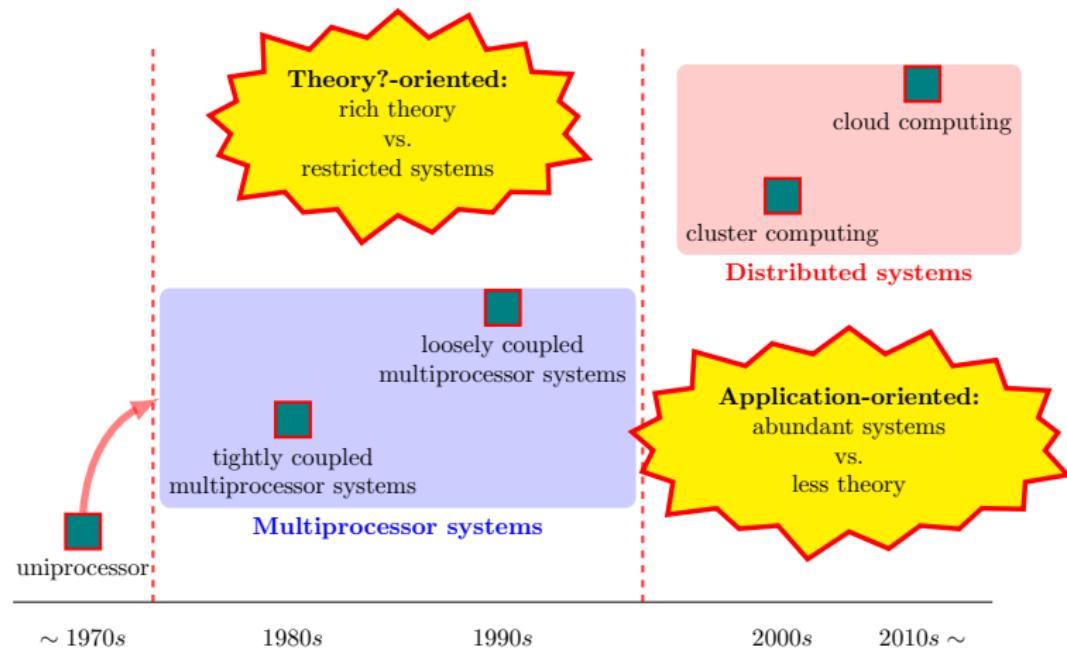
论文研究问题:

考虑到上述权衡，
面向大规模分布式系统的
分布数据一致性理论应具有什么特性？

考察分布数据一致性问题研究的历史：

- ▶ 核心权衡与解决方案
- ▶ 理论与系统

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)

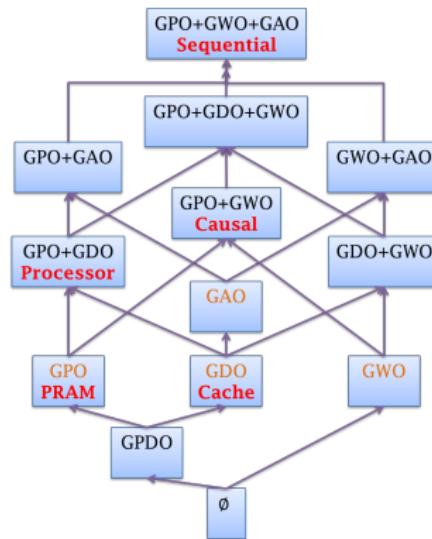
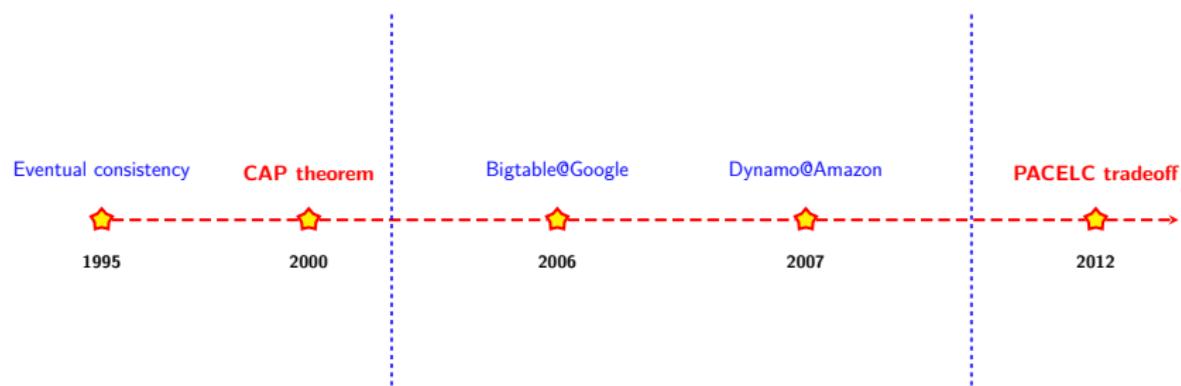


图: 一致性模型 (依据 [Steinke@JACM'04] 中 Figure 13 重绘).

核心权衡: 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

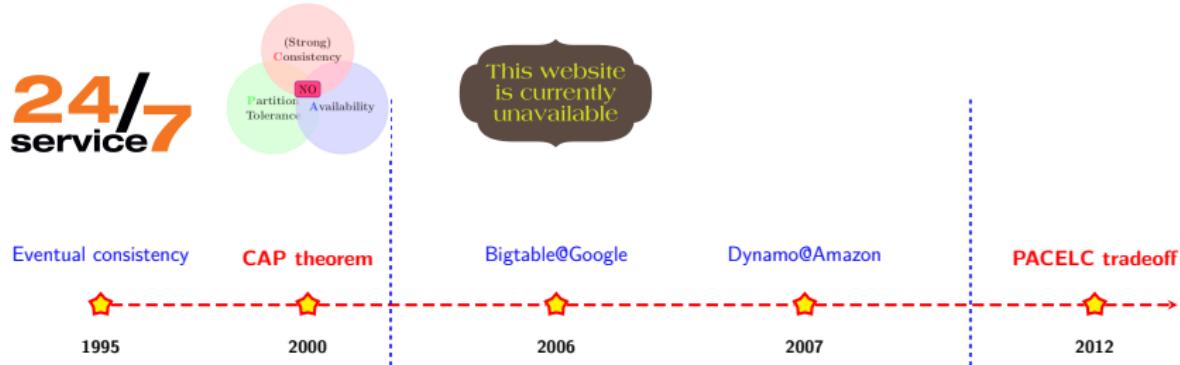
Douglas B. Terry, Marvin M. Thurber, Kara Barnes, Alan J. Demers,
Mike J. Spreitzer and Carl H. Hauser

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

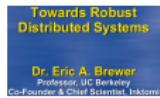
数据一致性问题研究的历史阶段 (分布式系统)



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

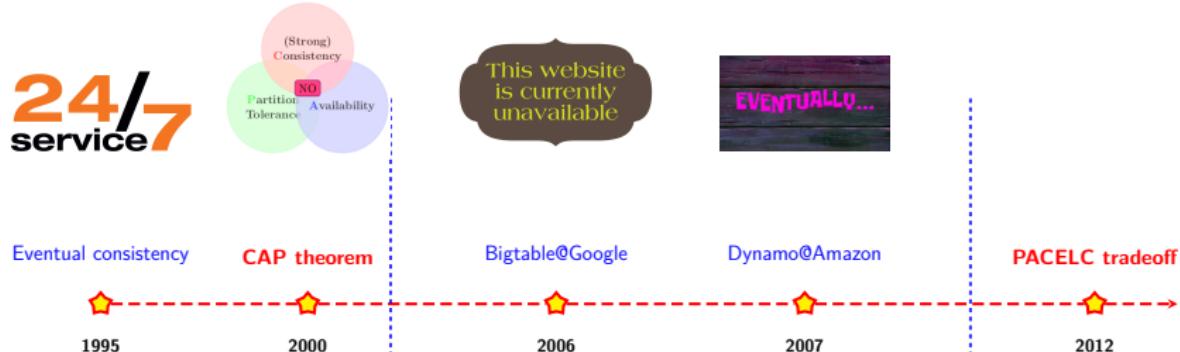
Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304 U.S.A.



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.

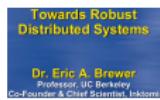
数据一致性问题研究的历史阶段 (分布式系统)



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

Douglas B. Terry, Marvin M. Thurber, Kara 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. Heid, David A. Hoffman, Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Ketukapat, Arvind Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

[fay,jeff,sanjay,wilson,csmith,dhoff,giuseppe,deniz,madan,jampani,gunavardhan,arvind,alex,peter,werner]@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 and Werner Vogels

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.

Towards Robust
Distributed Systems

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

Bigtable: A Distributed Storage System for Structured Data

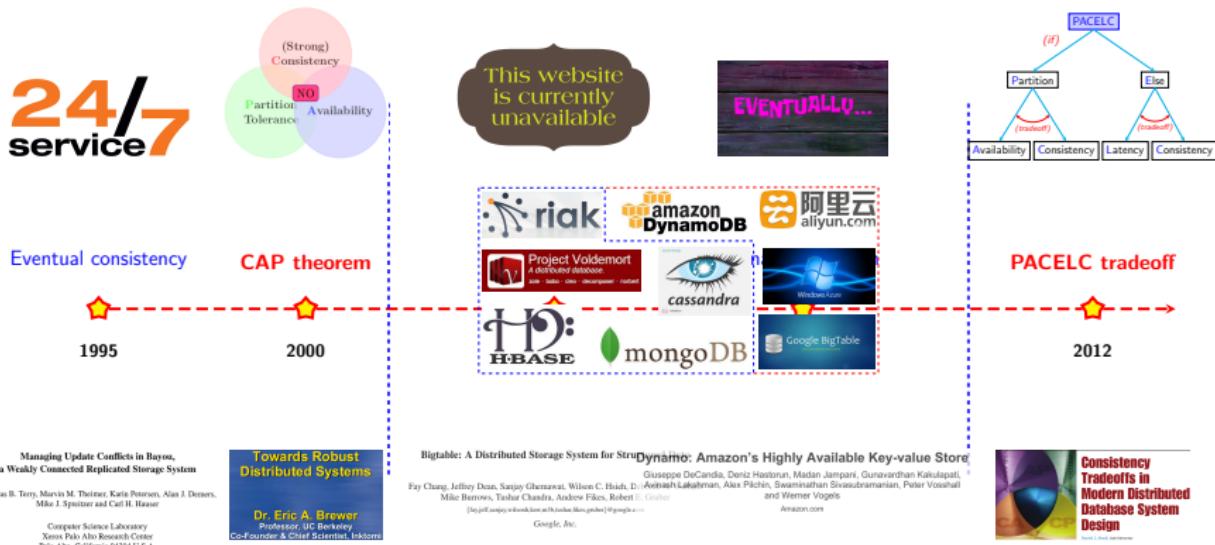
Dynamo: Amazon's Highly Available Key-value Store

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Heid, D. Arvind 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}@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？

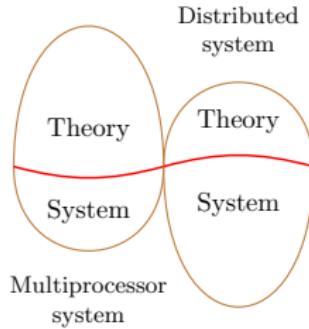
数据一致性问题研究的历史阶段 (结论)

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



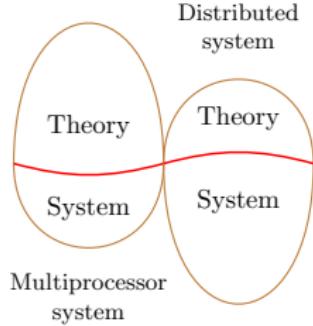
数据一致性问题研究的历史阶段 (结论)

新平台的两个特点:

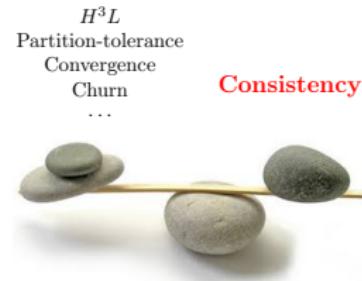
- (1) 云计算新平台凸显应用价值观
- (2) 应用价值观积极拥抱 tradeoffs

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

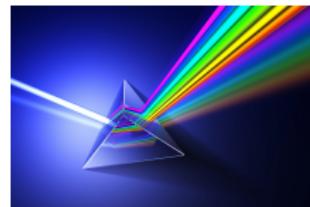


可调节: think *dynamically* [Terry@SOSP'13]

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)

论文研究问题:

如何在大规模分布式系统中
落实“多样化, 可调节; 精细化, 可度量”
这一体现应用价值观的数据一致性问题研究理念?

数据一致性问题研究的发展趋势及我们的工作 (III)

论文研究问题:

如何在大规模分布式系统中
落实“多样化, 可调节; 精细化, 可度量”
这一体现应用价值观的数据一致性问题研究理念?

论文主要贡献:

理念: 提出“以应用为导向的”、“多样化, 可调节; 精细化,
可度量”的一致性问题研究理念

数据一致性问题研究的发展趋势及我们的工作 (III)

论文研究问题:

如何在大规模分布式系统中
落实“多样化, 可调节; 精细化, 可度量”
这一体现应用价值观的数据一致性问题研究理念?

论文主要贡献:

理念: 提出“以应用为导向的”、“多样化, 可调节; 精细化,
可度量”的一致性问题研究理念

VPC:

数据一致性问题研究的发展趋势及我们的工作 (III)

论文研究问题:

如何在大规模分布式系统中
落实“多样化, 可调节; 精细化, 可度量”
这一体现应用价值观的数据一致性问题研究理念?

论文主要贡献:

理念: 提出“以应用为导向的”、“多样化, 可调节; 精细化,
可度量”的一致性问题研究理念

VPC:

PA2AM:

数据一致性问题研究的发展趋势及我们的工作 (III)

论文研究问题:

如何在大规模分布式系统中
落实“多样化, 可调节; 精细化, 可度量”
这一体现应用价值观的数据一致性问题研究理念?

论文主要贡献:

理念: 提出“以应用为导向的”、“多样化, 可调节; 精细化,
可度量”的一致性问题研究理念

VPC:

PA2AM:

RVSI:

分布数据一致性技术研究

1 研究背景

2 研究问题

3 相关工作

4 技术框架

相关工作分类

表：“多样化, 可调节; 精细化, 可度量”研究理念相关工作.

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”					
“精细化, 可度量”	验证				
	量化				

相关工作分类

表：“多样化, 可调节; 精细化, 可度量”研究理念相关工作.

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”				软件 事务内存	
“精细化, 可度量”	验证				
	量化				

“多样化, 可调节” 的研究理念 (一)

	读写寄存器	事务	
	多处理器系统	分布式系统	多处理器系统
“多样化, 可调节”	✓		

“多样化, 可调节” 的研究理念 (一)

	读写寄存器	事务	
	多处理器系统	分布式系统	多处理器系统
“多样化, 可调节”	✓		

典型: Hybrid consistency [Attiya@SIAM J. Comput.'98]

思想: 将操作分为强弱两类

“多样化, 可调节” 的研究理念 (一)

	读写寄存器	事务	
	多处理器系统	分布式系统	多处理器系统
“多样化, 可调节”	✓		

典型: Hybrid consistency [Attiya@SIAM J. Comput.'98]

思想: 将操作分为强弱两类

其它: “带同步的” 一致性模型 [Dubois@IEEE Computer'88] [Steinke@JACM'04]

特点: 强调正确性 (properly synchronized)

“多样化, 可调节” 的研究理念 (一)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”	相关工作丰富 理论扎实			

典型: Hybrid consistency [Attiya@SIAM J. Comput.'98]

思想: 将操作分为强弱两类

其它: “带同步的” 一致性模型 [Dubois@IEEE Computer'88] [Steinke@JACM'04]

特点: 强调正确性 (properly synchronized)

“多样化, 可调节” 的研究理念 (二)

	读写寄存器		事务	
	多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”	相关工作丰富 理论扎实	✓		

“多样化, 可调节” 的研究理念 (二)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”	相关工作丰富 理论扎实	✓		

思想: 借鉴并发展 Hybrid consistency 的思想

- 典型:
- ▶ Causal+forced+immediate operations [Ladin@TOCS'92]
 - ▶ RedBlue consistency [Li@OSDI'12]
 - ▶ Pileus [Terry@SOSP'13]

“多样化, 可调节” 的研究理念 (二)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”	相关工作丰富 理论扎实	✓		

思想: 借鉴并发展 Hybrid consistency 的思想

- 典型:**
- ▶ Causal+forced+immediate operations [Ladin@TOCS'92]
 - ▶ RedBlue consistency [Li@OSDI'12]
 - ▶ Pileus [Terry@SOSP'13]

特点: 更细粒度的多一致性模型共存、更能容忍数据不一致

“多样化, 可调节” 的研究理念 (二)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化, 可调节”	相关工作丰富 理论扎实	渐成趋势 理论欠缺		

思想: 借鉴并发展 Hybrid consistency 的思想

典型:

- ▶ Causal+forced+immediate operations [Ladin@TOCS'92]
- ▶ RedBlue consistency [Li@OSDI'12]
- ▶ Pileus [Terry@SOSP'13]

特点: 更细粒度的多一致性模型共存、更能容忍数据不一致

“多样化, 可调节” 的研究理念 (三)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	
多样化, 可调节”	相关工作丰富 理论扎实	渐成趋势 理论欠缺	软件事务内存	✓

思想: 多个事务一致性模型共存

- 典型:
- ▶ RC-SR (relaxed currency serializability) [Bernstein@SIGMOD'06]
 - ▶ Pileus consistency choices [Terry@MSR-TR'13]
 - ▶ Multi-level CSI (Causal Snapshot Isolation) [Tripathi@BigData'15]

“多样化, 可调节” 的研究理念 (三)

	读写寄存器	事务		
	多处理器系统	分布式系统	多处理器系统	
多样化, 可调节”	相关工作丰富 理论扎实	渐成趋势 理论欠缺	软件事务内存	✓

思想: 多个事务一致性模型共存

- 典型:
- ▶ RC-SR (relaxed currency serializability) [Bernstein@SIGMOD'06]
 - ▶ Pileus consistency choices [Terry@MSR-TR'13]
 - ▶ Multi-level CSI (Causal Snapshot Isolation) [Tripathi@BigData'15]

挑战: “多样化” 事务语义; 可扩展的系统实现

“多样化, 可调节” 的研究理念 (三)

	读写寄存器	事务	
	多处理器系统	分布式系统	多处理器系统
多样化, 可调节”	相关工作丰富 理论扎实	渐成趋势 理论欠缺	软件事务内存

思想: 多个事务一致性模型共存

典型:

- ▶ RC-SR (relaxed currency serializability) [Bernstein@SIGMOD'06]
- ▶ Pileus consistency choices [Terry@MSR-TR'13]
- ▶ Multi-level CSI (Causal Snapshot Isolation) [Tripathi@BigData'15]

挑战: “多样化” 事务语义; 可扩展的系统实现

“精细化, 可度量”的研究理念 (一)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证				
	量化				

“精细化, 可度量”的研究理念 (一)

		读写寄存器	事务		
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	✓			
	量化				

典型的一致性模型验证 (Verify) 问题:

- ▶ VSC (Sequential Consistency), VL (Linearizability) [Gibbons@SIAM J. Comput.'97]
- ▶ VMC (Memory Coherence) [Cantin@SPAA'03] [Cantin@TPDS'05]
- ▶ VTSO (Total Store Order) [Hangal@ISCA'03] [Manovit@SPAA'05] [Roy@CAV'06] [Baswana@CAV'08]

“精细化, 可度量”的研究理念 (一)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面			
	量化				

典型的一致性模型验证 (Verify) 问题:

- ▶ VSC (Sequential Consistency), VL (Linearizability) [Gibbons@SIAM J. Comput.'97]
- ▶ VMC (Memory Coherence) [Cantin@SPAA'03] [Cantin@TPDS'05]
- ▶ VTSO (Total Store Order) [Hangal@ISCA'03] [Manovit@SPAA'05] [Roy@CAV'06] [Baswana@CAV'08]

“精细化, 可度量”的研究理念 (一)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面			
	量化	暂无 强调正确性			

典型的一致性模型验证 (Verify) 问题:

- ▶ VSC (Sequential Consistency), VL (Linearizability) [Gibbons@SIAM J. Comput.'97]
- ▶ VMC (Memory Coherence) [Cantin@SPAA'03] [Cantin@TPDS'05]
- ▶ VTSO (Total Store Order) [Hangal@ISCA'03] [Manovit@SPAA'05] [Roy@CAV'06] [Baswana@CAV'08]

“精细化, 可度量”的研究理念 (二)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	✓		
	量化	暂无 强调正确性			

动机: 商业条款 SLA (Service Level Agreement) [Amazon@SOSP'07]

“精细化, 可度量”的研究理念 (二)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	✓		
	量化	暂无 强调正确性			

动机: 商业条款 SLA (Service Level Agreement) [Amazon@SOSP'07]

特点: 在线验证 safeness, regularity, atomicity [Golab@PODC'11]

“精细化, 可度量”的研究理念 (二)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究		
	量化	暂无 强调正确性			

动机: 商业条款 SLA (Service Level Agreement) [Amazon@SOSP'07]

特点: 在线验证 safeness, regularity, atomicity [Golab@PODC'11]

不足: 常用 Pipelined-RAM consistency, causal consistency, hybrid consistency 验证问题有待研究

“精细化, 可度量”的研究理念 (三)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究		
	量化	暂无 强调正确性			

“精细化, 可度量”的研究理念 (三)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究		
	量化	暂无 强调正确性	✓		

量化执行: $k/\Delta/\Gamma$ -atomicity [Golab@PODC'11, ICDCS'13, ICDCS'14, PODC'15]

量化协议: probabilistic regularity/atomicity

[Yu@DISC'03] [Lee@DC'05] [Gramoli@OPODIS'07] [Bailis@PVLDB'12]

“精细化, 可度量”的研究理念 (三)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究		
	量化	暂无 强调正确性	量化执行易 量化协议难		

量化执行: $k/\Delta/\Gamma$ -atomicity [Golab@PODC'11, ICDCS'13, ICDCS'14, PODC'15]

量化协议: probabilistic regularity/atomicity

[Yu@DISC'03] [Lee@DC'05] [Gramoli@OPODIS'07] [Bailis@PVLDB'12]

“精细化, 可度量”的研究理念 (四)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究	软件事务内存	✓
	量化	暂无 强调正确性	量化执行易 量化协议难		

▶ SR (Serializability) 强一致性模型及变体

[Papadimitriou@JACM'79] [Bernstein@TODS'83] [Yannakakis@JACM'84]

▶ SI (Snapshot Isolation) 等弱一致性模型

[Adya@Phd-Thesis'99] [Fekete@TODS'05] [Cahill@SIGMOD'08] [Zellag@VLDB'14]

“精细化, 可度量”的研究理念 (四)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究	软件事务内存	✓
	量化	暂无 强调正确性	量化执行易 量化协议难		

▶ SR (Serializability) 强一致性模型及变体

[Papadimitriou@JACM'79] [Bernstein@TODS'83] [Yannakakis@JACM'84]

▶ SI (Snapshot Isolation) 等弱一致性模型

[Adya@Phd-Thesis'99] [Fekete@TODS'05] [Cahill@SIGMOD'08] [Zellag@VLDB'14]

“精细化, 可度量”的研究理念 (四)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究	软件事务内存	理论全面 指导协议设计
	量化	暂无 强调正确性	量化执行易 量化协议难		

▶ SR (Serializability) 强一致性模型及变体

[Papadimitriou@JACM'79] [Bernstein@TODS'83] [Yannakakis@JACM'84]

▶ SI (Snapshot Isolation) 等弱一致性模型

[Adya@Phd-Thesis'99] [Fekete@TODS'05] [Cahill@SIGMOD'08] [Zellag@VLDB'14]

“精细化, 可度量”的研究理念 (五)

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“精细化, 可度量”	验证	典型模型 理论全面	弱模型验证 有待研究	软件事务内存	理论全面 指导协议设计
	量化	暂无 强调正确性	量化执行易 量化协议难		量化协议难 相关工作少

- ▶ 量化 SI (Snapshot Isolation) 与 RC (Read Committed) 协议 [Fekete@VLDB'09]

相关工作总结

表：“多样化，可调节；精细化，可度量”研究理念相关工作.

		读写寄存器		事务	
		多处理器系统	分布式系统	多处理器系统	分布式系统
“多样化，可调节”		相关工作丰富 理论扎实	渐成趋势 理论欠缺	软件 事务内存	探索阶段
“精细化，可度量”	验证	典型模型 理论全面	弱模型验证 有待研究		理论全面 指导协议设计
	量化	暂无 强调正确性	量化执行易 量化协议难		量化协议难 相关工作少

分布数据一致性技术研究

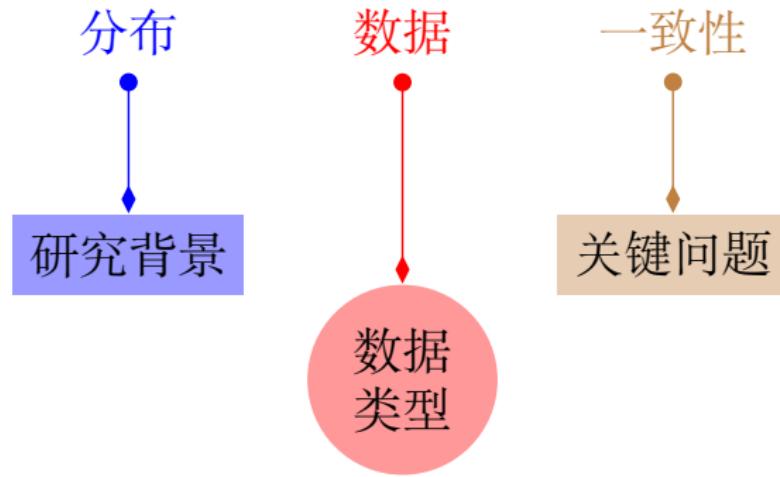
1 研究背景

2 研究问题

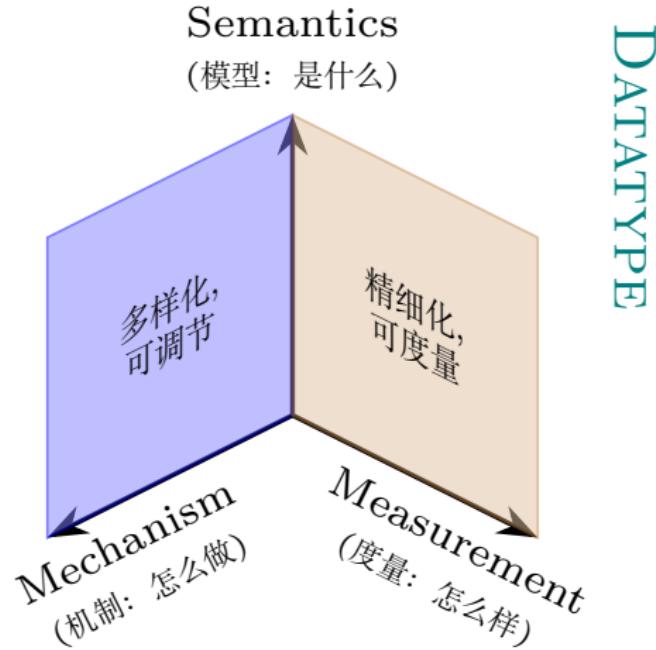
3 相关工作

4 技术框架

分布数据一致性问题



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

- ▶ 读写寄存器

数据类型

数据类型：从个体到群组

- ▶ 读写寄存器

- ▶ 事务对象

- ▶ 事务 \triangleq 多个读写寄存器的操作序列

- ▶ 支持“all-or-none”语义

- ▶ 易于开发并发应用

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

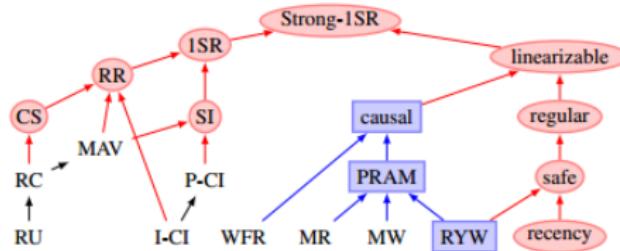


图: 一致性模型强弱关系 (来自 [Bailis@VLDB'14]).

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



“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