

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

July 23, 2016

分布数据一致性理论与技术研究

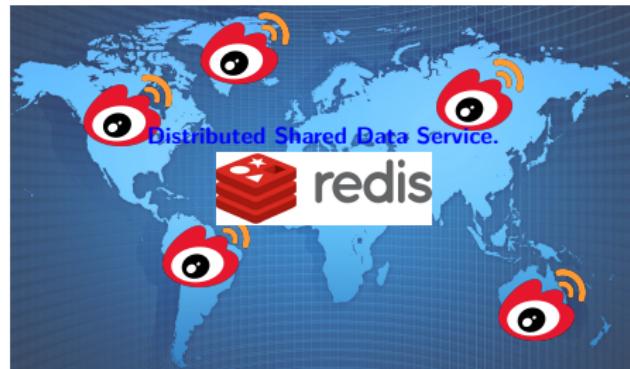
① 研究背景

② 研究问题

③ 研究方法

④ 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

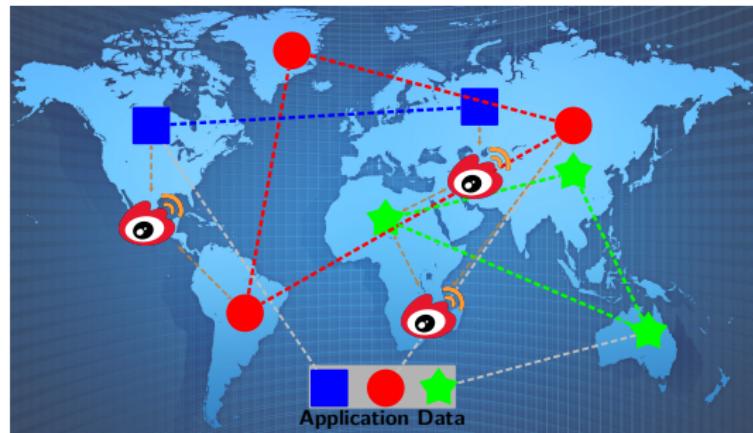
底层数据服务系统特性需求:

- ▶ 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.

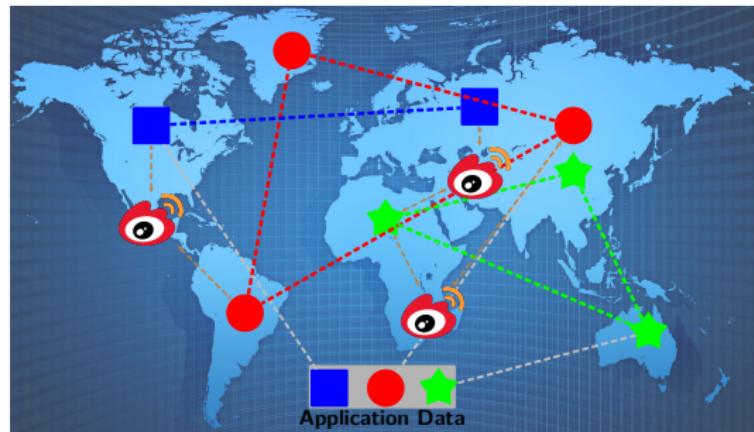
² 数据来自 InfoQ.

分布数据



应用数据:

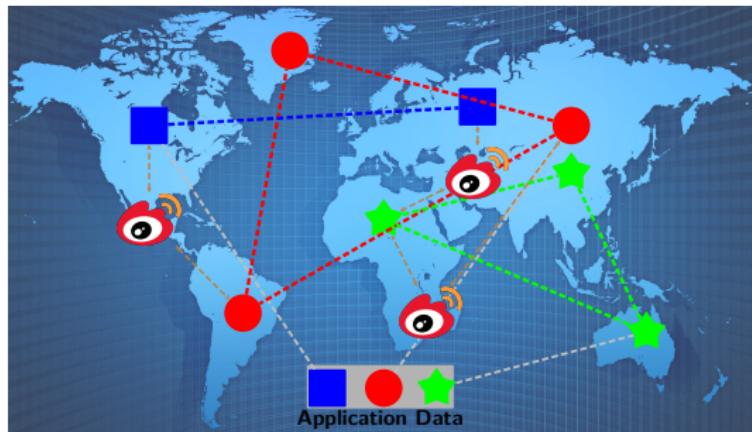
分布数据



应用数据:

1. 分区 (partition): 水平扩展

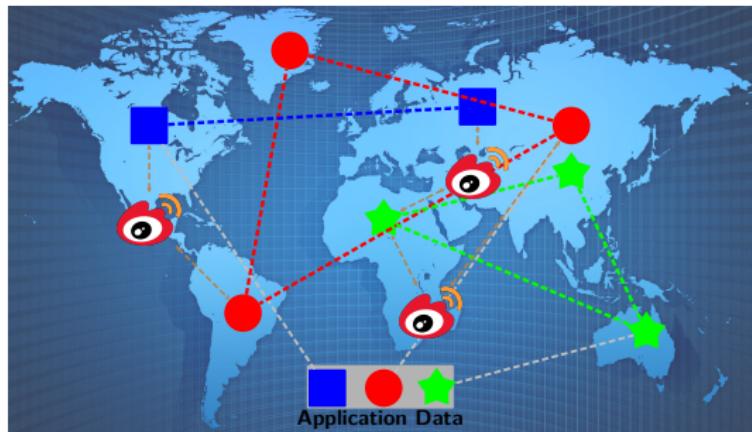
分布数据



应用数据:

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

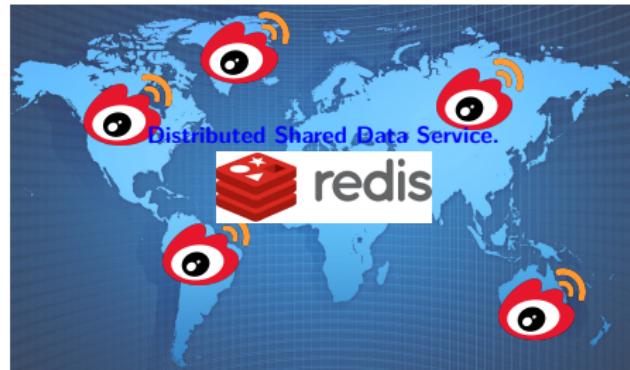
分布数据



分布数据:

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):

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)

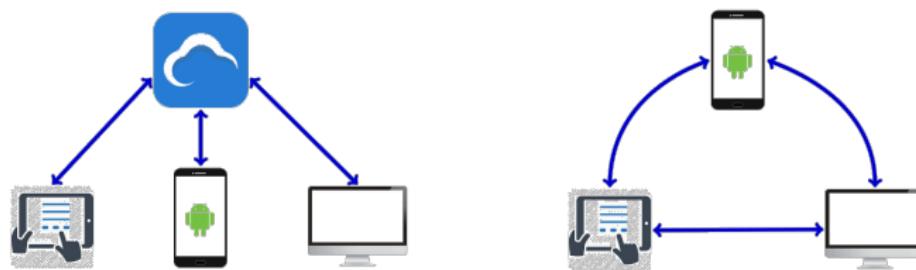


图: 个人多设备文件共享 ([基于云] C/S 结构 [左] & P2P 结构 [右]).

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

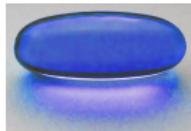
1 研究背景

2 研究问题

3 研究方法

4 未来工作

分布数据一致性问题

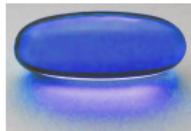


理想情况 (The Blue Pill):

- ▶ 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 (The Blue Pill):

- ▶ 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 (The Red Pill):



分布数据一致性是分布共享
数据服务的核心问题

分布数据一致性问题举例 (I)

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.

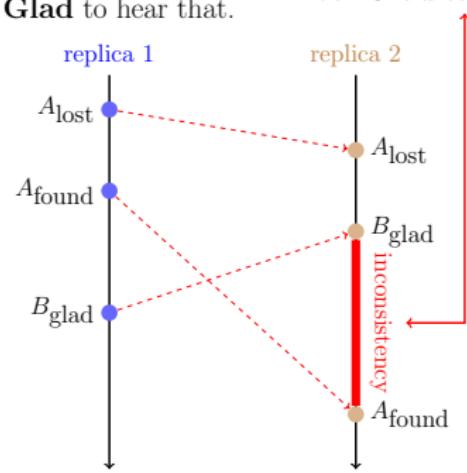


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

分布数据一致性问题举例 (II)



图: 多设备文件共享时, 更新丢失 ($\#N = 3, \#W = 2, \#R = 1$).

分布数据一致性问题举例 (II)

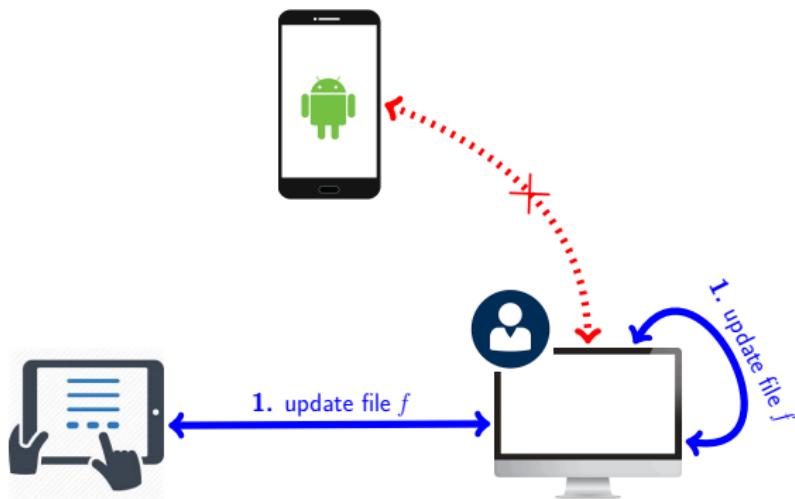


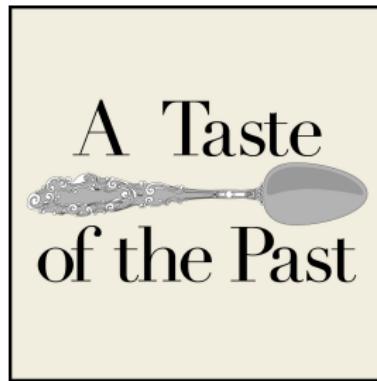
图: 多设备文件共享时, 更新丢失 ($\#N = 3, \#W = 2, \#R = 1$).

分布数据一致性问题举例 (II)



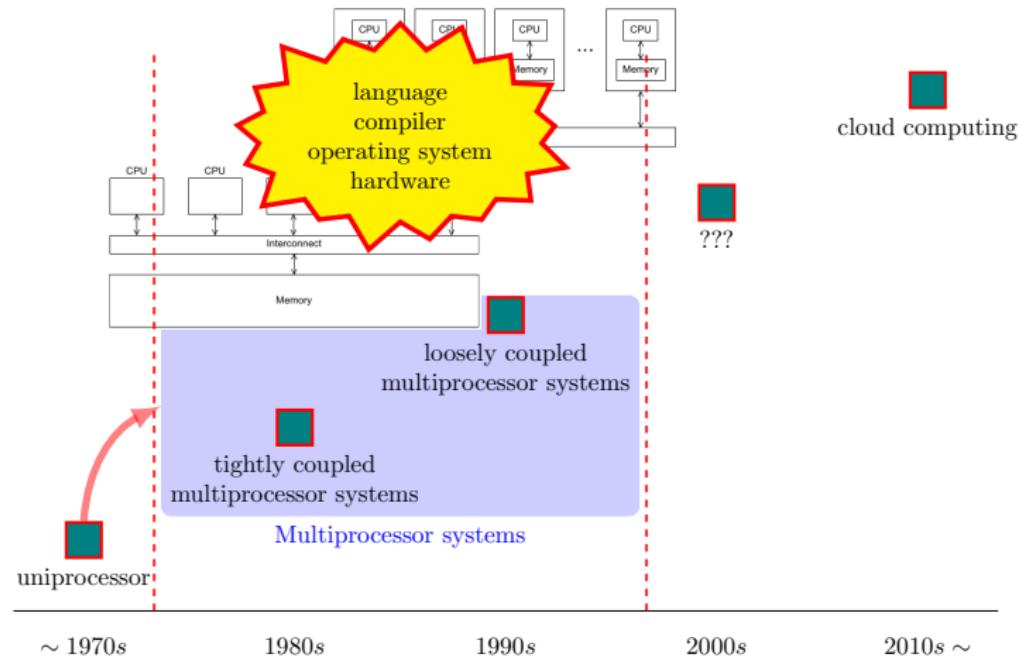
图: 多设备文件共享时, 更新丢失 ($\#N = 3, \#W = 2, \#R = 1$).

数据一致性问题研究的历史阶段



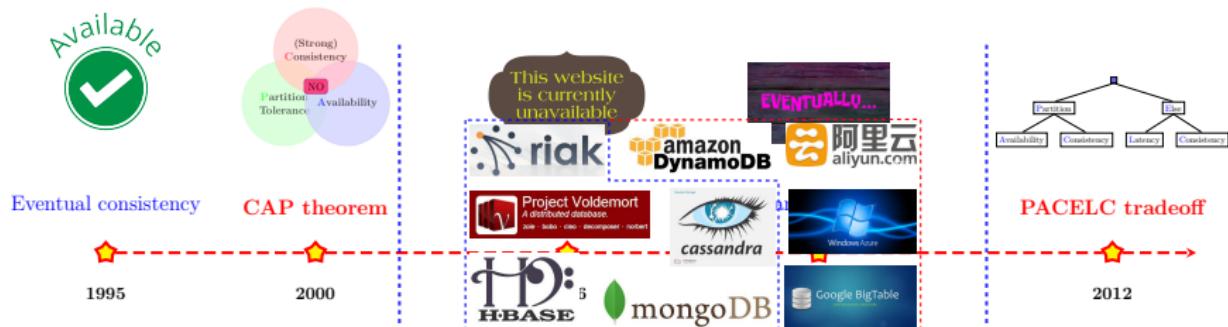
1. 理论 vs. 系统
2. 以一致性为核心的 tradeoffs

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)

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. Theimer, Katie Poosman, 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

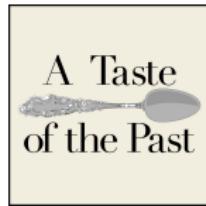
Dynamo: Amazon's Highly Available Key-value Store
Giuseppe DeCandia, Deniz Hastorun, Madan Jannotti, Gunavardhan Kekulapati,
Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Heinz, David Iyengar, Avinash Lakshman, Alex Pilchin,
Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

Amazon.com

Consistency
Tradeoffs in
Modern Distributed
Database System
Design

数据一致性问题研究的历史阶段

theory vs. system (egg)



数据一致性问题研究的发展趋势

云计算凸显应用价值观
应用价值观拥抱 tradeoffs



数据一致性问题研究的发展趋势

云计算凸显应用价值观
应用价值观拥抱 tradeoffs



购物车 SLA [Terry@SOSP'13]:

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

数据一致性问题研究的发展趋势

云计算凸显应用价值观
应用价值观拥抱 tradeoffs



购物车 SLA [Terry@SOSP'13]:

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

出租车实时位置查询 SLA:

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

数据一致性问题研究的发展趋势 (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 (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 (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

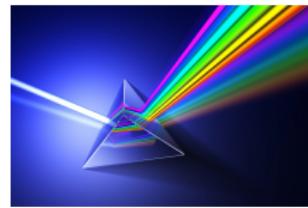


可调节: think *dynamically* [Terry@SOSP'13]

依据应用需求/系统状态调节数据一致性

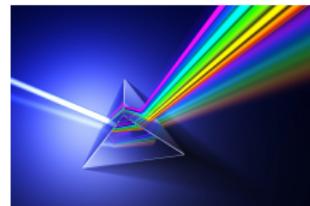
数据一致性问题研究的发展趋势 (I)

精细化：从二元到连续谱



数据一致性问题研究的发展趋势 (I)

精细化: 从二元到连续谱



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



量化系统执行, 后验系统对一致性的满足程度

我们的工作

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 未来工作

分布数据一致性理论与技术研究

1 研究背景

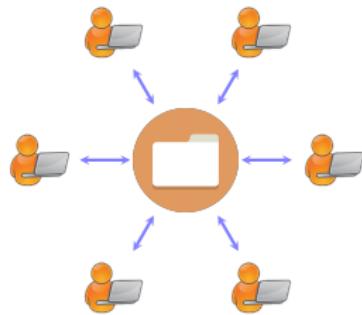
2 研究问题

3 研究方法

- 理论模型: 分布共享数据
- 技术框架

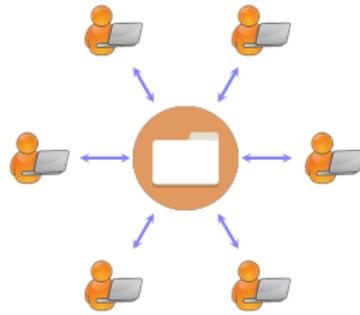
4 未来工作

分布共享数据模型

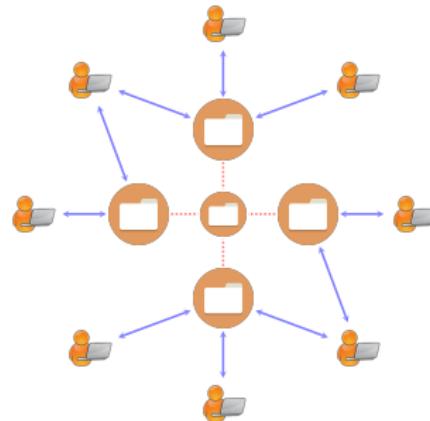


共享数据系统 (single copy)

分布共享数据模型

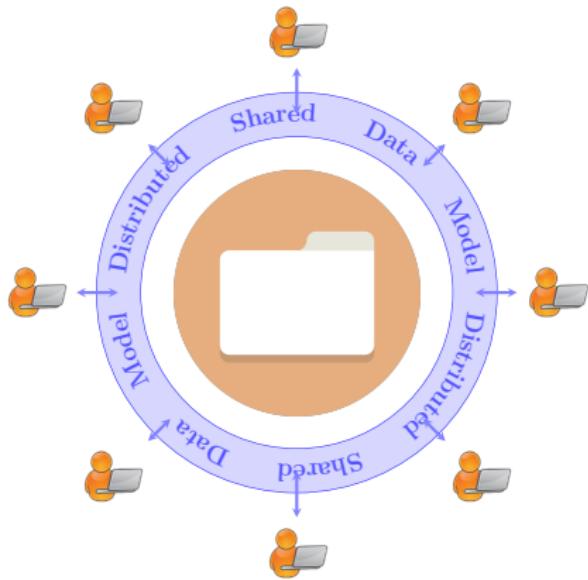


共享数据系统 (single copy)



分布数据系统 (replicas)

分布共享数据模型



分布共享数据模型: 在分布数据之上提供共享数据的假象

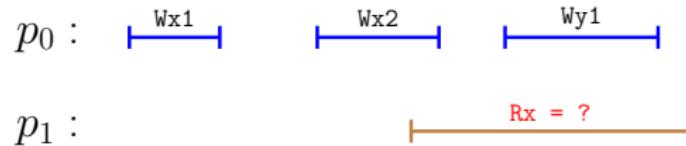
分布共享数据模型

p_0, p_1 : 客户进程 x, y : 共享变量

分布共享数据模型

p_0, p_1 : 客户进程 x, y : 共享变量

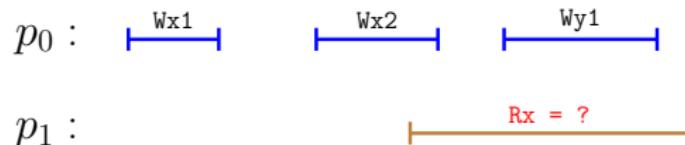
多进程并发读/写共享变量:



分布共享数据模型

p_0, p_1 : 客户进程 x, y : 共享变量

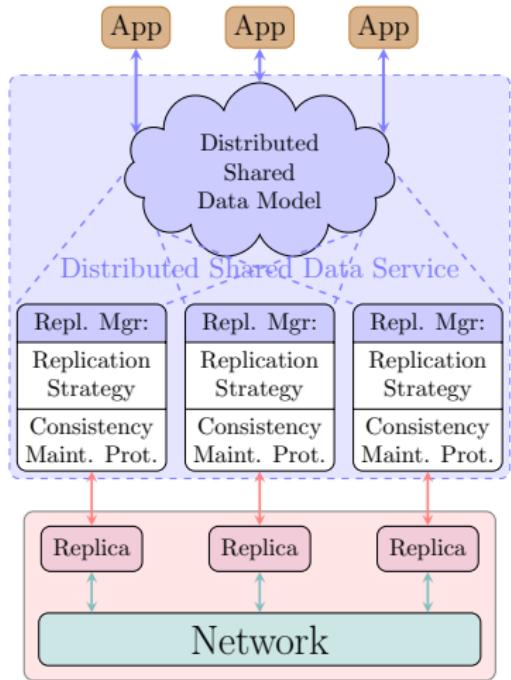
多进程并发读/写共享变量:



数据一致性问题: 读操作允许返回什么值?

不同一致性模型 $\xrightarrow[\text{定义}]{\text{规定}}$ 不同合法返回值

分布共享数据服务



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. 精细化, 可度量
2. 多样化, 可调节

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 未来工作

分布数据一致性问题

分布数据一致性问题：

- ✓ 分布: partition + replication
- ✗ 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: partition + replication
- ✗ 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: partition + replication
- ✗ 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:

- ▶ 语义 (semantics; WHAT)
- ▶ 机制 (mechanisms; HOW)
- ▶ 度量 (measurements; ???)

技术框架

TODO: 数据类型 + 一致性关键问题技术框架

数据类型

数据类型：从个体到群组

- ▶ 单独读写变量 (*read/write registers*)



Key-Value Store

数据类型

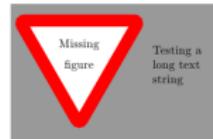
数据类型：从个体到群组

- ▶ 单独读写变量 (*read/write registers*)
- ▶ 事务对象 (*transactions*)
 - ▶ 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



cassandra

Key-Value Store



一致性模型

一致性模型 [Steinke@JACM'04]:

- ▶ 规定读操作 (*read*) 所允许的返回值

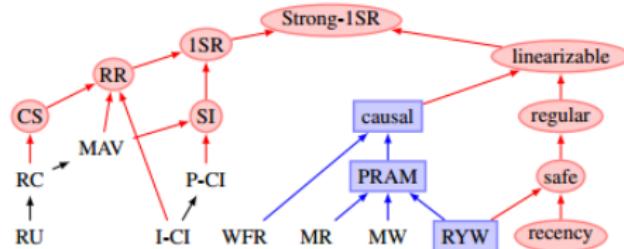


图: 来自 [Bailis@VLDB'14]

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} , 设计维护算法及系统
 \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} , 设计维护算法及系统
 \mathcal{S} :

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

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



“多样化, 可调节” 的挑战:

- ▶ 兼容的混合一致性模型
- ▶ 手段之一: 参数化



一致性度量方法

给定分布式系统 S 及某一致性模型 \mathcal{C} ,
度量数据一致性:

一致性度量方法

给定分布式系统 \mathcal{S} 及某一致性模型 \mathcal{C} ,
度量数据一致性:

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

给定分布式系统 \mathcal{S} 及某一致性模型 \mathcal{C} ,
度量数据一致性:

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”的挑战:

- ▶ 问题复杂度分析, 算法设计
- ▶ 数学建模与分析

"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 未来工作

分布数据一致性问题研究的发展趋势 (I)

分布数据一致性问题研究的发展趋势 (II)