

分布共享数据服务理论与技术研究

魏恒峰

导师: 吕建 黄宇

南京大学软件所

July 4, 2016

分布共享数据服务理论与技术研究

1 研究背景

2 研究问题

3 研究方法

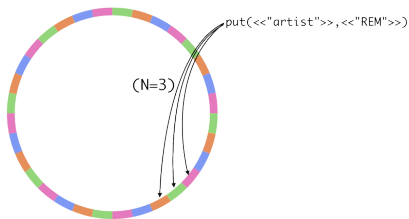
分布式应用



TODO: 动画: 分布部署

分布数据

(**TODO:** 动画: partition + replication)



distributed data : partition & replication

分布数据典型应用 (I)



图: 分布式存储系统 (开源 [左] & 商用 [右]).

应用需求 [Facebook@OSDI'10] vs. “分布数据”:

低延迟: 就近访问副本数据

高可用性, 高容错性: 备份容灾

分布数据典型应用 (II)

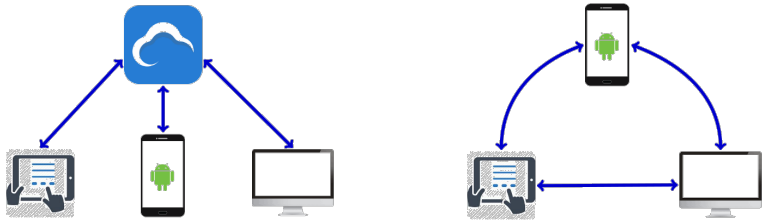


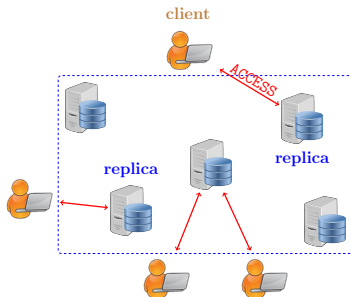
图: 个人多设备文件共享 ([基于云] C/S 结构 [左] & P2P 结构 [右]).

应用需求与特点 [Strauss@MIT Thesis'10] vs. “分布数据”:

功能需求: 文件副本

网络断连: 备份容灾; 离线可用

分布式应用访问分布数据



TODO: 重绘: 显示 partition & replication

分布共享数据服务

- ▶ 简化上层应用的开发

TODO: 图: 分布共享数据服务

分布共享数据服务

TODO: 图: 分布共享数据服务

- ▶ 简化上层应用的开发
- ▶ 提供共享数据的抽象

分布共享数据服务

TODO: 图: 分布共享数据服务

- ▶ 简化上层应用的开发
- ▶ 提供共享数据的抽象
- ▶ 屏蔽底层数据的分布性

分布共享数据服务理论与技术研究

1 研究背景

2 研究问题

3 研究方法

面向共享数据的编程模型

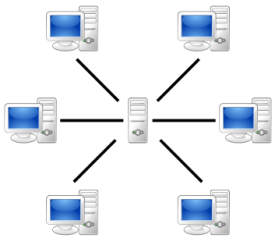


图: (集中式) 共享数据系统 **TODO: 重绘.**

共享变量编程模型 (**TODO: happy programmer**):

- ▶ 数据层: 数据建模为变量
- ▶ 业务层: 读—计算—写
- ▶ 变量全局唯一 (one-copy)
- ▶ 隐含假设: 变量随时可访问
- ▶ 读操作返回最新值

面向分布数据的编程模型 (I)

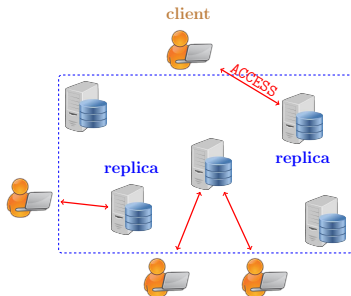


图: 分布数据系统.

共享变量编程模型与分布数据不匹配:

- ▶ 数据副本 (replication)
- ▶ 没有全局唯一变量的概念
- ▶ 节点/通讯故障
- ▶ 读操作语义无定义

面向分布数据的编程模型 (II)

消息传递编程模型:

- ▶ 读/写 + 通信 (communication)
- ▶ 从哪里读
- ▶ 写哪里去
- ▶ 如何理解返回值
- ▶ 如何处理失败

面向分布数据的编程模型 (II)

消息传递编程模型:

- ▶ 读/写 + 通信 (communication)
- ▶ 从哪里读
- ▶ 写哪里去
- ▶ 如何理解返回值
- ▶ 如何处理失败

消息传递编程模型的缺点: (TODO: unhappy programmer)

数据层: 暴露分布数据细节

业务层: 难于编程

数据层: 数据语义不明确

业务层: 难于保证程序正确性

面向分布数据的编程模型 (III)

共享变量编程模型与分布数据不匹配的根本原因:
分布数据一致性问题

定义 (数据一致性问题 (非形式化定义))

读操作语义问题: 在分布数据环境下, 读操作允许返回什么值?

面向分布数据的编程模型 (III)

共享变量编程模型与分布数据不匹配的根本原因:
分布数据一致性问题

定义 (数据一致性问题 (非形式化定义))

读操作语义问题: 在分布数据环境下, 读操作允许返回什么值?

如何理解数据一致性问题:

1. 数据层数据一致与否是相对于业务层逻辑而言的
2. 可通过业务层出现的异常行为 (anomalies) 来衡量数据是否一致

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.

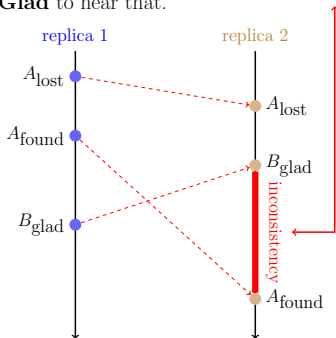


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

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)

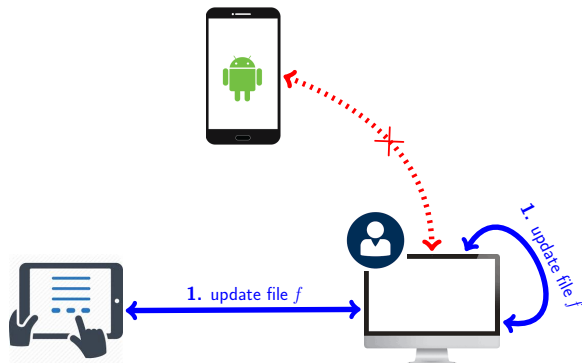


图: 多设备文件共享时, 更新丢失 ($\#N = 3, \#W = 2, \#R = 1$).

数据一致性问题举例 (II)

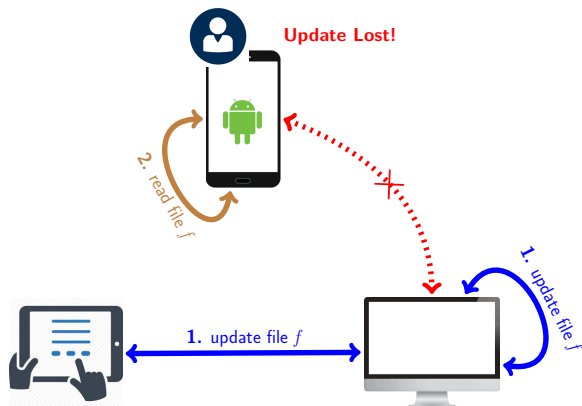


图: 多设备文件共享时, 更新丢失 ($\#N = 3, \#W = 2, \#R = 1$).

面向分布数据的编程模型 (IV)

共享变量编程模型与分布数据不匹配: 数据一致性问题

消息传递编程模型: 业务层显式处理数据一致性问题

- ▶ 理论难度大, 实现繁琐
- ▶ 处理不当: 异常

面向分布数据的编程模型 (V)

问题:

如何解决共享变量编程模型与分布数据不匹配的问题?

分布共享数据服务: 在**分布数据**之上提供**共享数据**的假象

面向分布数据的编程模型 (V)

问题:

如何解决共享变量编程模型与分布数据不匹配的问题?

分布共享数据服务: 在**分布数据**之上提供**共享数据**的假象

业务层: 采用“自然的”共享变量编程模型

数据层: 分布数据

面向分布数据的编程模型 (V)

问题:

如何解决共享变量编程模型与分布数据不匹配的问题?

分布共享数据服务: 在**分布数据**之上提供**共享数据**的假象

业务层: 采用“自然的”共享变量编程模型

分布共享数据服务: 专注分布数据一致性问题, 实现分布数据透明性

- ▶ 屏蔽消息传递编程模型的通信细节
- ▶ 形式化定义读操作语义

数据层: 分布数据

分布共享数据服务应用举例 (I)

```
// Alice's post
put(key = lost_msg, val = "lost")
put_after(key = found_msg, val = "found", dep = lost_msg)

// Bob's reply
get(key = found_msg) // get "found"
put(key = glad_msg, val = "glad", dep = found_msg)

// Eve's view
read(key = glad_msg) // get "glad"
read(key = found_msg) // get "found" instead of "NULL"
```

Listing 1: 在社交网络应用中使用满足“因果一致性”的共享变量.

分布共享数据服务应用举例 (II)

Another example: coordination???

分布共享数据服务理论与技术研究

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
- 技术途径: 三维框架

分布共享数据服务理论与技术研究

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
- 技术途径: 三维框架

分布共享数据 (I)

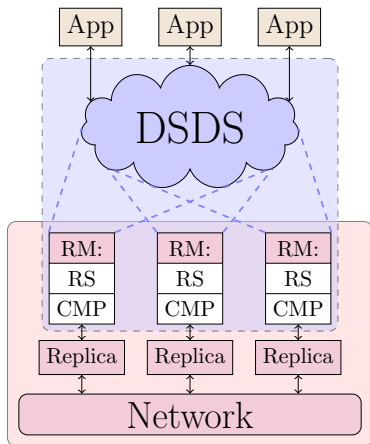
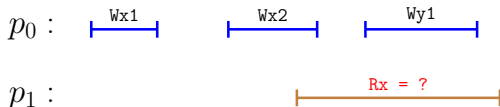


图: 分布数据共享服务.

分布共享数据 (II)

x, y : 共享变量 p_0, p_1 : 客户进程

多进程并发提交 (读/写) 操作:



问题: 读操作允许返回什么值?

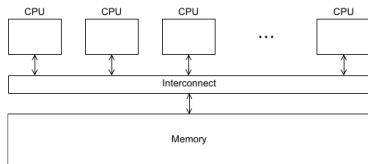
不同一致性 $\xrightarrow[\text{定义}]{\text{规定}}$ 不同合法返回值

分布共享数据 (III)

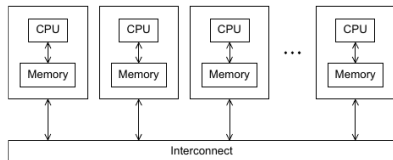
基本定位: 传统概念应用于新型平台

分布共享数据服务: 分布共享内存模型 + 分布数据系统

分布共享内存 (I)



(a) 共享内存系统.



(b) 分布内存系统.

图: 多处理器系统体系结构 **TODO: 重绘.**

分布共享内存 (II)

分布内存系统实例: 按系统耦合度分类

分布共享内存 (III)

分布共享内存: 在分布内存之上提供共享内存的假象

TODO: 图: 分布共享内存 (from Kai Li)

分布共享数据 (IV)

问题空间: 传统问题, 新平台, 新挑战 (**TODO: 总结**)

目的: 并行编程

实现手段: 硬件/操作系统

分布共享数据服务理论与技术研究

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
- 技术途径: 三维框架

分布共享内存中的数据一致性问题

数据一致性问题的三个层面:

- 1. 虚拟共享数据有什么? ▶ 数据类型
- 2. 上层接口语义是什么? ▶ 一致性模型
- 3. 底层消息传递为什么? ▶ 一致性保障

研究框架

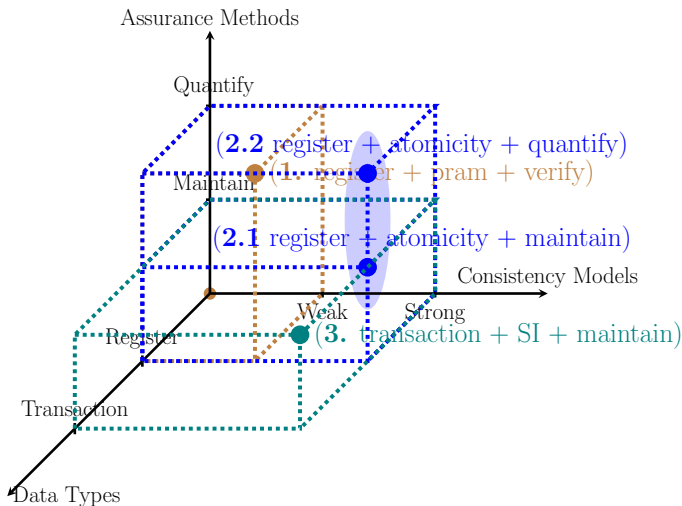


图: 数据一致性及保障技术研究框架

研究挑战



用户对一致性的需求:

1. 多样化, 可定制 [Terry@CACM'13]

2. 精细化, 可度量 [Bailis@VLDB'12]

研究挑战



用户对一致性的需求:

1. 多样化, 可定制 [Terry@CACM'13]

多样化:

- ▶ 一致性族: causality; read-your-writes (RYW)
- ▶ 参数调节: 提供“有限度”的不一致 [Yu@TOCS'02]

可定制: 混合使用, 运行时可变

2. 精细化, 可度量 [Bailis@VLDB'12]

研究挑战



用户对一致性的需求:

1. 多样化, 可定制 [Terry@CACM'13]

- 多样化:
- ▶ 一致性族: causality; read-your-writes (RYW)
 - ▶ 参数调节: 提供“有限度”的不一致 [Yu@TOCS'02]

可定制: 混合使用, 运行时可变

2. 精细化, 可度量 [Bailis@VLDB'12]

精细化: “在大多数情况下, 访问到一致数据”

可度量: 量化系统执行, 后验系统对一致性的满足程度