

# 分布复制数据一致性及保障技术研究

魏恒峰

导师: 吕建 黄宇

南京大学软件所

May 11, 2016

# 分布复制数据一致性及保障技术研究

1 研究背景

2 研究问题

3 研究方法

# 分布式应用

开放互联的网络环境, 软件/应用体现出“分布性”: 地理分布部署

**TODO:** 图: 分布式应用

# 数据复制

## 数据共享

**TODO: 图示: 集中式  $\Rightarrow$  数据复制**

集中式数据:

- ▶ 性能瓶颈
- ▶ 单点故障

数据复制:

- ▶ 负载均衡, 就近访问
- ▶ 容灾备份

# 分布副本数据的典型应用 (一)



图: 分布式存储系统 (开源 [左] & 商用 [右])

上层业务需求 [Facebook@OSDI'10] vs. “副本技术”:

高性能: 低延迟就近访问副本数据

高可靠性: 备份容灾; 随时可用

## 分布数据的典型应用 (二)

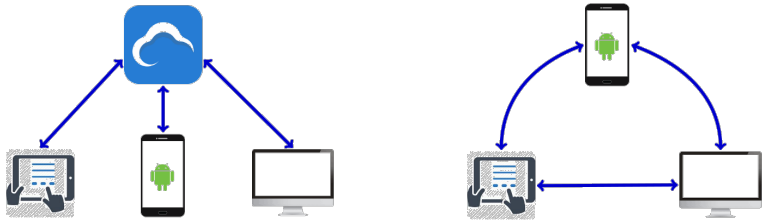


图: 个人多设备文件共享 ([基于云] C/S 结构 [左] & P2P 结构 [右])

应用需求与特点 [Strauss@MIT Thesis'10] vs. “副本技术”:

功能需求: 文件副本

网络断连: 容灾备份; 离线可用

# 数据的形态 (总结)

分布共享数据常以**副本**的形态存在:

功能需求: 如, 多设备文件共享

系统性能: 低延迟就近访问

高可靠性: 备份容灾; 避免单点故障

# 分布复制数据一致性及保障技术研究

1 研究背景

2 研究问题

3 研究方法



# Programming with Replicated Data

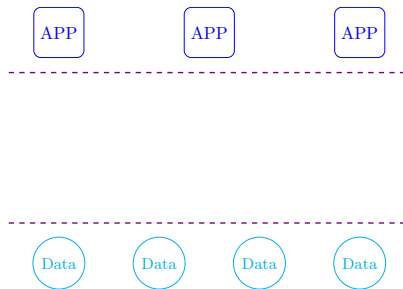
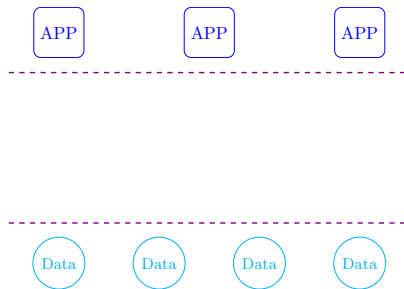


图: 数据副本导致编程抽象失配 **TODO: 重绘.**

# Programming with Replicated Data



开发人员: 读—计算—写

- ▶ 集中的“变量”视角
- ▶ 顺序访问  $\Rightarrow$  顺序语义
- ▶ 隐含假设: 随时可用

图: 数据副本导致编程抽象失配 **TODO: 重绘.**

# Programming with Replicated Data



开发人员: 读—计算—写

- ▶ 集中的“变量”视角
- ▶ 顺序访问  $\Rightarrow$  顺序语义
- ▶ 隐含假设: 随时可用



图: 数据副本导致编程抽象失配 **TODO: 重绘.**

现实: **TODO: 总结**

- ▶ 副本数据
- ▶ 并发访问
- ▶ 节点/通讯故障

# Programming with Replicated Data



图: 数据副本导致编程抽象失配 **TODO: 重绘.**

通信 (Communication):

- ▶ 从哪里读
- ▶ 写哪里去
- ▶ 失败如何处理

# 数据一致性问题

编程抽象失配  $\Rightarrow$  数据一致性问题:

- ▶ 异常 (anomalies)
- ▶ 难于编程
- ▶ 难于保证正确性

# 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.

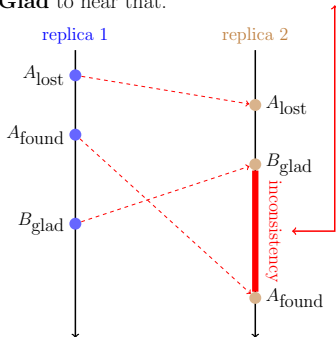


图: 社交网络中, 消息-评论乱序 [Lloyd@CACM'14].

# 数据一致性问题



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

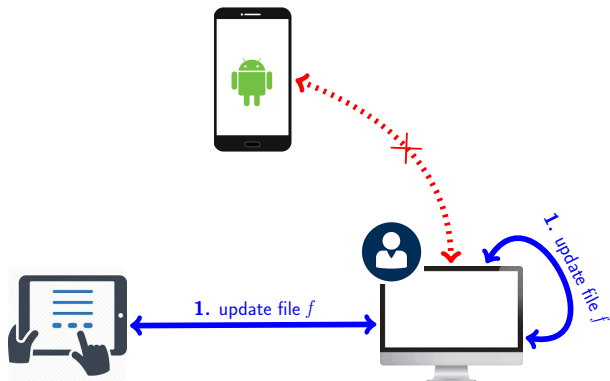


图: 多设备文件共享时, 更新丢失 ( $\#N = 3, \#W = 2, \#R = 1$ ).



# 数据一致性问题

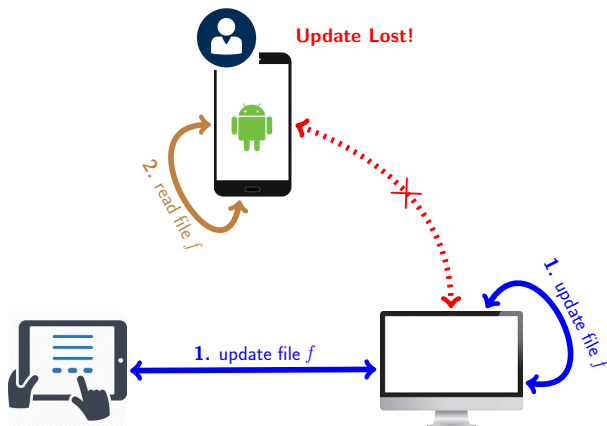


图: 多设备文件共享时, 更新丢失 ( $\#N = 3, \#W = 2, \#R = 1$ ).

# DSM Service

**TODO:** 图: DSM Service

针对数据一致性问题:

- ▶ 提供共享变量抽象
- ▶ 提供并发语义
- ▶ 屏蔽节点/通讯故障

# 分布复制数据一致性及保障技术研究

- 1 研究背景
- 2 研究问题
- 3 研究方法

