



清华大学  
Tsinghua University

# 存储系统研究的一些体会与经验

汪庆

清华大学计算机系

wq1997@tsinghua.edu.cn

## 汪庆 — 清华存储实验室博士后

- ❖ 2023年6月于清华大学获博士学位（导师：舒继武教授）
- ❖ 研究方向 — 分布式存储系统：**存储**(内存、持久性内存) + **网络**(交换机、网卡)
- ❖ 以第一作者发表了OSDI、FAST、SIGMOD等国际会议论文
- ❖ 参与了实验室**10余项**研究工作
- ❖ 主页：<http://storage.cs.tsinghua.edu.cn/~wq/>



清华存储实验室



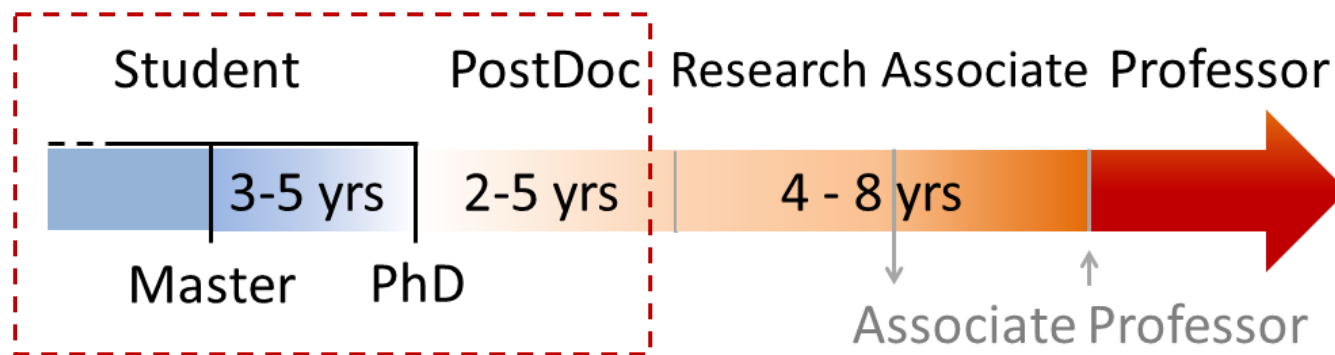
舒继武 教授



陆游游 副教授

## 报告的局限性

- ❖ 刚毕业博士的观点，而非成熟科研人员的观点
- ❖ 聚焦于单个研究工作的流程，而非更宏大的（例如研究路线）
- ❖ 个人的体会，而非普适的结论





# 大纲

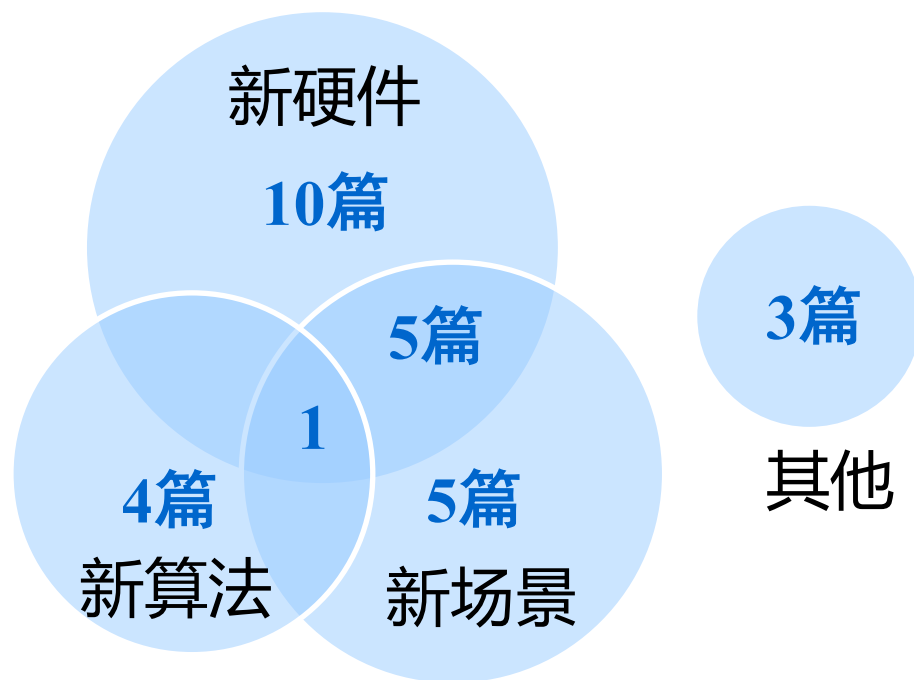
1 研究想法的产生、细化与验证

2 科研论文的写作

3 总结

## 存储系统软件发展的几种驱动力

- ❖ **新硬件**：持久性内存、可编程网络器件、CXL、UPMEM
- ❖ **新算法与技术**：learned index、新纠删算法、新形式化方法、新编译技术
- ❖ **新场景与需求**：AI、serverless、云原生、fail slow、GDPR



## 找研究话题 (2)

为什么是现在？而不是十年前、上世纪？

❖ 新硬件？

❖ 例如：可编程交换机，可以降低分布式存储系统的协调开销

❖ 新算法与技术？

❖ 例如：eBPF技术的普及，加速、自定义OS的存储模块

❖ 新场景与需求？

❖ 例如：云原生场景，需考虑存储系统的弹性

具体、有特色的研究话题

Orderless and Eventually Durable File Systems

A Client-Centric Approach to Transactional Datastores

The Dennis M. Ritchie Award中存储相关的博士论文名



高性能的LSM系统



低尾延迟的LSM系统

## 研究想法（Idea）的核心是定义研究问题

- ❖ 如果研究问题新颖，简单的解法即可
  - ❖ 例如：容忍fail slow的副本协议 [Copilot, OSDI@20]
- ❖ 如果研究问题常见，需要进一步定义
  - ❖ 例如：多核文件系统
  - ❖ 进一步定义1：适用不同崩溃一致性方法(journaling、log-structured...)的多核文件系统架构
  - ❖ 进一步定义2：硬件事务内存(HTM)加速的多核文件系统
  - ❖ 进一步定义的过程中，包含着**1.和现有工作的区分(对现有工作的分析)**，**2.技术创新**

## 找研究想法 ( 2 )





回答一个问题：**为什么别人不做？**

可能的答案：

- ❖ 1. 别人做过或正在做
- ❖ 2. 因为某种原因，只有我想到了

## 答案1：别人做过或正在做

### 如何知道别人做过了什么：

- ❖ 阅读，持续地阅读
- ❖ 所有的会议论文、每天的arXiv
- ❖ 当开了会才去了解论文内容，就已经晚了
- ❖ 会议的Accept List，向作者邮件询问

Accepted papers

—

We have 40 papers accepted tentatively for SoCC 2023. Paper title, content, and even the acceptance decision may change during the shepherding process. The detailed program is coming up soon.

Research

—

Plexus: Optimizing Join Approximation for Geo-Distributed Data Analytics  
- Joel Wolfrath, Abhishek Chandra (University of Minnesota)

Carbon Containers: A System-level Facility for Managing Application-level Carbon Emissions  
- John Thiede (University of Massachusetts - Amherst); Noman Bashir (University of Massachusetts Amherst); David Irwin (University of Massachusetts, Amherst); Prashant Shenoy

SOCC 2023接收论文列表

Accepted Papers

The following papers have been accepted to appear at the 29th ACM SIGOPS Symposium on Operating Systems Principles (SOSP), conditional on the approval of each paper's shepherd:

**A Cloud-Scale Characterization of Remote Procedure Calls** by Korakit Seemakhupt (University of Virginia), Brent E. Stephens (Google and University of Utah), Samira Khan (Google and University of Virginia), Sihang Liu (University of Waterloo), Hassan Wassel (Google), Soheil Hassas Yeganeh (Google), Alex C. Snoeren (Google and UC San Diego), Arvind Krishnamurthy (Google and University of Washington), David Culler (Google) and Henry M. Levy (Google and University of Washington)

**Acto: Automatic End-to-End Testing for Operation Correctness of Cloud System Management** by Jiawei Tyler Gu (University of Illinois at Urbana-Champaign), Xudong Sun (University of Illinois at Urbana-Champaign), Wentao Zhang (University of Illinois at Urbana-Champaign), Yuxuan Jiang (University of Illinois at Urbana-Champaign), Chen Wang (IBM Research), Mandana Vaziri (IBM Research), Owlolabi Legunsen (Cornell University) and Tianyin Xu (University of Illinois at Urbana-Champaign)

**Antipode: Enforcing Cross-Service Causal Consistency in Distributed Applications** by João Loff (INESC-ID, Instituto Superior Técnico, Universidade de Lisboa), Daniel Porto (INESC-ID, Instituto Superior Técnico, Universidade de Lisboa), João Garcia (INESC-ID, Instituto Superior Técnico, Universidade de Lisboa), Jonathan Mace (Max Planck Institute for Software Systems and Microsoft Research) and Rodrigo Rodrigues (INESC-ID, Instituto Superior Técnico, Universidade de Lisboa)

SOSP 2023接收论文列表

## 答案1：别人做过或正在做

### 如何知道别人正在做什么：

- ❖ 关注：很多研究者乐于在主页分享自己的研究进展
- ❖ 交流：与相同研究方向的研究者多讨论

#### About me













CS PhD student at Columbia University working on eBPF applications with professor [Asaf Cidon](#).  
Check out our [XRP project](#) which accelerates storage accesses using eBPF.

#### Current projects:

- Accelerating NVMeoF with XRP.
- Using eBPF for configurable page cache eviction.
- Accelerating a networked key-value store using eBPF.

Previously, I was a software engineer at [Arrikto](#) working on cloud-native machine learning infrastructure ([Kubeflow](#)). I also created the [Rook Cassandra Operator](#) and the [Scylla Operator](#), which was adopted and continued by the company as their official solution to deploy on Kubernetes.

## XRP作者的主页

	<a href="#">Aleksandar Dragojevic</a>	[已发送] Re: ... [EXTERNAL] Questions about FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" ...FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" I've had a look at the code and the keys are actually fixed size. Sorry for the confusion, but I...	2020-01-03
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: ... [EXTERNAL] Questions about FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" ...FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" Thank you very much for replies :) I still have questions about variable-length keys support. I...	2020-01-03
	<a href="#">Aleksandar Dragojevic</a>	[已发送] Re: ... [EXTERNAL] Questions about FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" ...FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" Hi Qing Wang, Thank you for your interest in our work. Here are the answers to your questio...	2020-01-03
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: [EXTERNAL] Questions about FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" ...FaRM reads return consistent objects. Scans are made consistent using the clock mechanism. Some applications might not require consistent scans, so those don't hav...	2020-01-03
	<a href="#">alekd</a>	[已发送] Questions about FaRM B-Tree in paper "Fast General Distributed Transactions with Opacity" ...FaRM's B-Tree in MOTIVATION section. 1) The B-Tree nodes are distributed across different machines, rather sharded in different machines by key range. What confused...	2020-01-03
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM is a cool distributed system :) In some recent research papers (see 1,2,3), the researchers focus on the load imbalance between RPC connections (request processi...	2019-04-10
	<a href="#">Aleksandar Dragojevic</a>	[已发送] Re: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM is a cool distributed system :) In some recent research papers (see 1,2,3), the researchers focus on the load imbalance between RPC connections (request processi...	2019-04-10
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM have some mechanism to balance threads in a machine? Thanks and regards. Qing Wang Department of Computer Science and Technology Tsinghua University --...	2019-04-10
	<a href="#">Aleksandar Dragojevic</a>	[已发送] Re: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM have some mechanism to balance threads in a machine? Thanks and regards. Qing Wang Department of Computer Science and Technology Tsinghua University --...	2019-04-10
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM: Fast Remote Memory" , there we have a log per thread. Each thread keeps updates in a per-thread buffer stored in NVM. When the buffer is full, it gets sent to th...	2019-04-09
	<a href="#">Aleksandar Dragojevic</a>	[已发送] Re: ... Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM: Fast Remote Memory" , there we have a log per thread. Each thread keeps updates in a per-thread buffer stored in NVM. When the buffer is full, it gets sent to th...	2019-04-09
	<a href="#">Aleksandar Dragojevic</a>	[研究] RE: Questions about "No compromises: distributed transactions with consistency, availability, and performance" ...FaRM: Fast Remote Memory" , there we have a log per thread. Each thread keeps updates in a per-thread buffer stored in NVM. When the buffer is full, it gets sent to th...	2019-04-09

## 与FaRM作者邮件讨论

# 研究想法是否值得做（2）

答案1：别人做过或正在做

如何知道别人正在做什么：

- ❖ 我们组的一个小工具
- ❖ 捕捉学术主页的更新

PageDog: Your registered pages have been changed

Hi,

Here are the modified pages from `system` track.

- SKKU Data Intensive Computing Lab: <http://dicl.skku.edu/publications.html>
- OrderLab publication: <https://orderlab.io/pubs.html>
- SymbioticLab (publication): <https://symbioticlab.org/publications/#>

Best regards.

[SKKU Data Intensive Computing Lab](#) :

SKKU DICL

DICL

Home Publications

SKKU Data Intensive Computing Lab

Publications

Mijin An, Jonghyeok Park, Tianzheng Wang, Beomseok Nam, and Sang-Won Lee

NV-SQL: Boosting OLTP Performance with Non-Volatile DIMMs,

To appear at 49th International Conference on Very Large Databases (VLDB 2023), Sep. 2023.

Tuan Anh Ngyuen, Hyeongjun Jeon, Daegyu Han, Duck-...Kim, Sungsoon Park, Jinkyu Jeong and Beomseok Nam

NVMe-Driven Lazy Cache Coherence for Immutable Data with NVMe over Fabrics

答案2：因为某种原因，只有我想到了

这个原因到底是什么？

- ❖ 一些观察（observations）
  - ❖ 比如数据访问的一些特点
- ❖ 一些跨领域的知识
  - ❖ 例如网络的拥塞控制 -> 存储的请求流量控制？
- ❖ 一些只有少数人可得的硬件
  - ❖ CXL SSD？
- ❖ 更聪明？（可能性不大）

# 研究想法是否值得做（4）

回答一个问题：为什么别人不做？

可能的答案：

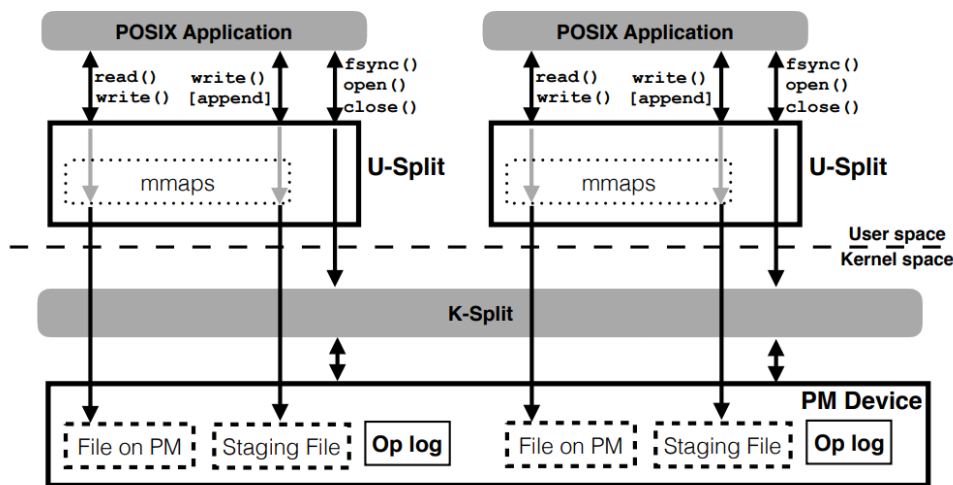
- ❖ 1. 别人做过或正在做
- ❖ 2. 因为某种原因，只有我想到了

这个问题的本质：

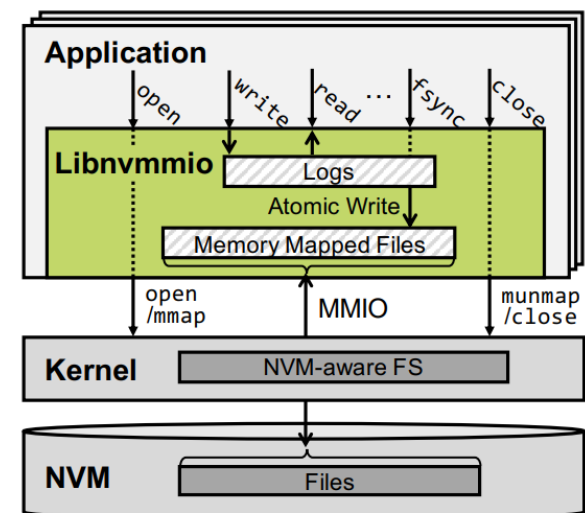
- ❖ 1. 相关工作的分析，定义scope
- ❖ 2. 创新性的分析，来自观察、跨领域知识还是硬件优势？

## 撞车了怎么办？

- ❖ 对于一些竞争性的Idea（例如最近的CXL、ZNS等），撞车是常态
- ❖ 工作的独特之处在哪儿？差异化
- ❖ 不可能有两个完全一样的研究工作，思考流程、着重点会有区别
- ❖ 顶会能容纳多个相似的工作，体现的是一种趋势



SplitFS (SOSP@19)



Libnvmio (ATC@20)

## 快速验证、减少沉没成本

- ❖ 不建议：花了大量时间把整个系统都实现了，然后发现idea没用？
- ❖ 建议：抽象问题，快速验证，验证有效后再构建系统
- ❖ 例子
  - ❖ Idea：冷/热数据分离加速存储系统
  - ❖ 涉及很多其他模块：冷热识别、并发控制、动态负载处理等
  - ❖ 简化与核心Idea无关的部分：假设冷热已知、数据集不变、不考虑并发正确性
  - ❖ 快速验证：手动地进行数据冷热分离，插入人工的判断语句，看是否性能提升
  - ❖ 若有效，才进行下一步：系统的实现

通过快速验证，相同时间可尝试更多的idea，容错性更大



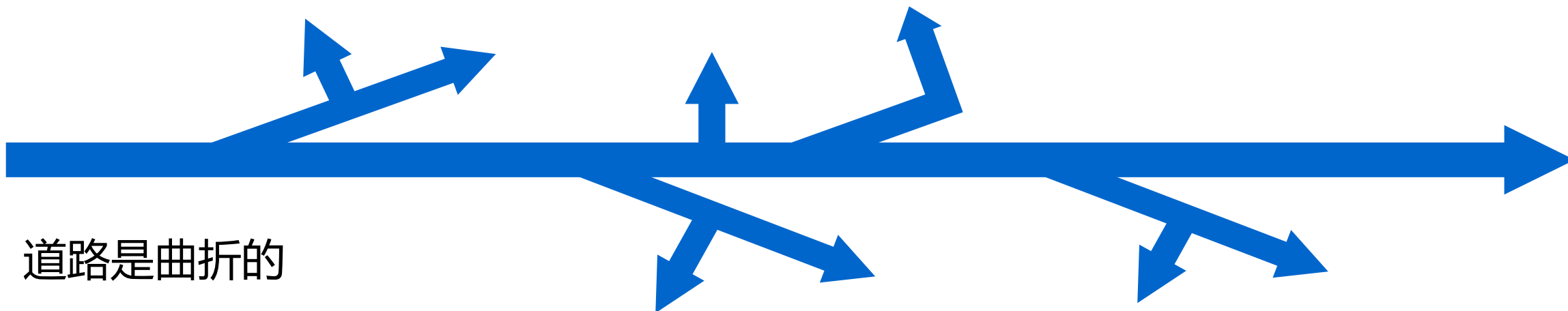
## 实现系统、做实验

- ❖ 时间较长（几个月到几年）
- ❖ 最重要的是**试错的经历**，升华丰富idea
  - ❖ idea实现的挑战在哪？
  - ❖ idea成立的条件是什么？
  - ❖ idea有效的原因是什么？

**Alternative designs.** We now discuss alternative designs to address item size variability, and why we do not adopt them.

1) *Use a dedicated set of machines to serve large requests*, as suggested in [45]. This solution may lead to waste of resources because the workloads of large and small requests cannot be consolidated. It also requires migrating items across machines in case an item changes size, and adds one network hop to redirect large requests.

2) *Splitting large operations in smaller chunks*. This allows interleaving the processing of such chunks with small requests. This design may lead to lower resource efficiency with respect to the run-to-completion model adopted by Minos. First, it may lead to worse data locality, by accessing memory regions corresponding to different requests, and



道路是曲折的



# 大纲

1 研究想法的产生、细化与验证

2 科研论文的写作

3 总结



# 什么时候开始构思论文？

## 什么时候开始构思论文？

- ❖ 写完了代码（做完了实验）后，然后构思文章框架，有几个问题：
  - ❖ 反推研究动机，不自然
  - ❖ 故事僵硬，新谄出来的，打磨的少
- ❖ **写第一行代码开始，就构思论文**（以PPT或草稿的形式）
- ❖ 从项目开始，讲上百次的故事，是否能吸引住别人（研究动机、问题、解法）？
- ❖ 在研究推进的过程中，根据实验结果逐步修正故事

**解释**、而非描述：“为什么这样设计” 比 “设计成什么样子” 更重要

## 描述性文本

先做了A、再做了B、然后C

实验报告式

## 解释性文本

先做了A，达到了什么效果

因为A能够XX，B操作更快

B完成了才能做C，因为一致性...

选择了C，没选择D，这是因为...

C虽然有优势，但存在...问题

体现思考，能让人获得insight



读者大脑的缓存有限，如何让里面都是核心、重要的内容？

- ❖ 提高重要观点的局部性（重复）
  - ❖ 适时地重复最重要的观点
- ❖ 减少随机访问（克制）
  - ❖ 减少名词的种类，例如Message、Packet、Request、Query、Operation
- ❖ 减少工作集大小（改变布局）
  - ❖ 一段话只讲一件事
  - ❖ 原来的布局：正常路径+各种corner cases混到一起写；
  - ❖ 重新布局：先写完正常路径；然后每个corner case一段段分开写

## 不存在完美的系统

- ❖ 例如存储软件系统里的RUM猜想（读性能、写性能、空间占用无法三者兼得）
- ❖ 研究工作的局限性在哪？不需要掩盖（会显得奇怪）
- ❖ 适时地阐明局限性（如何适时？讲完了优势之后？）

Mu has some limitations. First, Mu relies on RDMA and so it is suitable only for networks with RDMA, such as local area networks, but not across the wide area. Second, Mu is an in-memory system that does not persist data in stable storage—doing so would add additional latency dependent on the device speed.<sup>1</sup> However, we observe that the industry is working on extensions of RDMA for persistent memory, whereby RDMA writes can be flushed at a remote persistent memory with minimum latency [70]—once available, this extension will provide persistence for Mu.

Mu [OSDI@20]

Our approach has three limitations. First, NR incurs space overhead due to replication: it consumes  $n$  times more memory, where  $n$  is the number of nodes. Thus, NR is best suited for smaller structures that occupy just a fraction of the available memory (e.g., up to hundreds of MB). Second, NR is *blocking*: a thread that stops executing operations can block the progress of other threads; in practice, we did not find that to be a problem as long as threads keep executing operations on the data structure. Finding a non-blocking variant of NR is an interesting research direction. Finally, NR may

NR [ASPLOS@17, best paper]

## 计算机系统会议的审稿意见有时很苛刻

- ❖ 指出了几十个问题，包括创新性、写作、技术问题等
- ❖ 下次投稿时，若依照意见全部修改，导致论文面目全非，失去核心论点

## 客观评价审稿意见

- ❖ 哪些是共性问题？需要何种程度的修改？
- ❖ 哪些是无关紧要的问题（仅是评审人的个人风格）？不修改，避免影响论文风格
- ❖ 客观评价是找回自信的方式
- ❖ 不要被审稿人牵着鼻子走



## 科研想法：

- ❖ 多问几个问题：“为什么是现在？”、“为什么别人没做？”
- ❖ 快速验证、记录试错经历（论文素材）

## 论文写作：

- ❖ 项目一开始就构思
- ❖ 解释性文本、维护读者缓存
- ❖ 不遮盖局限性、客观评价审稿者意见

# 谢谢！

欢迎一起交流！

