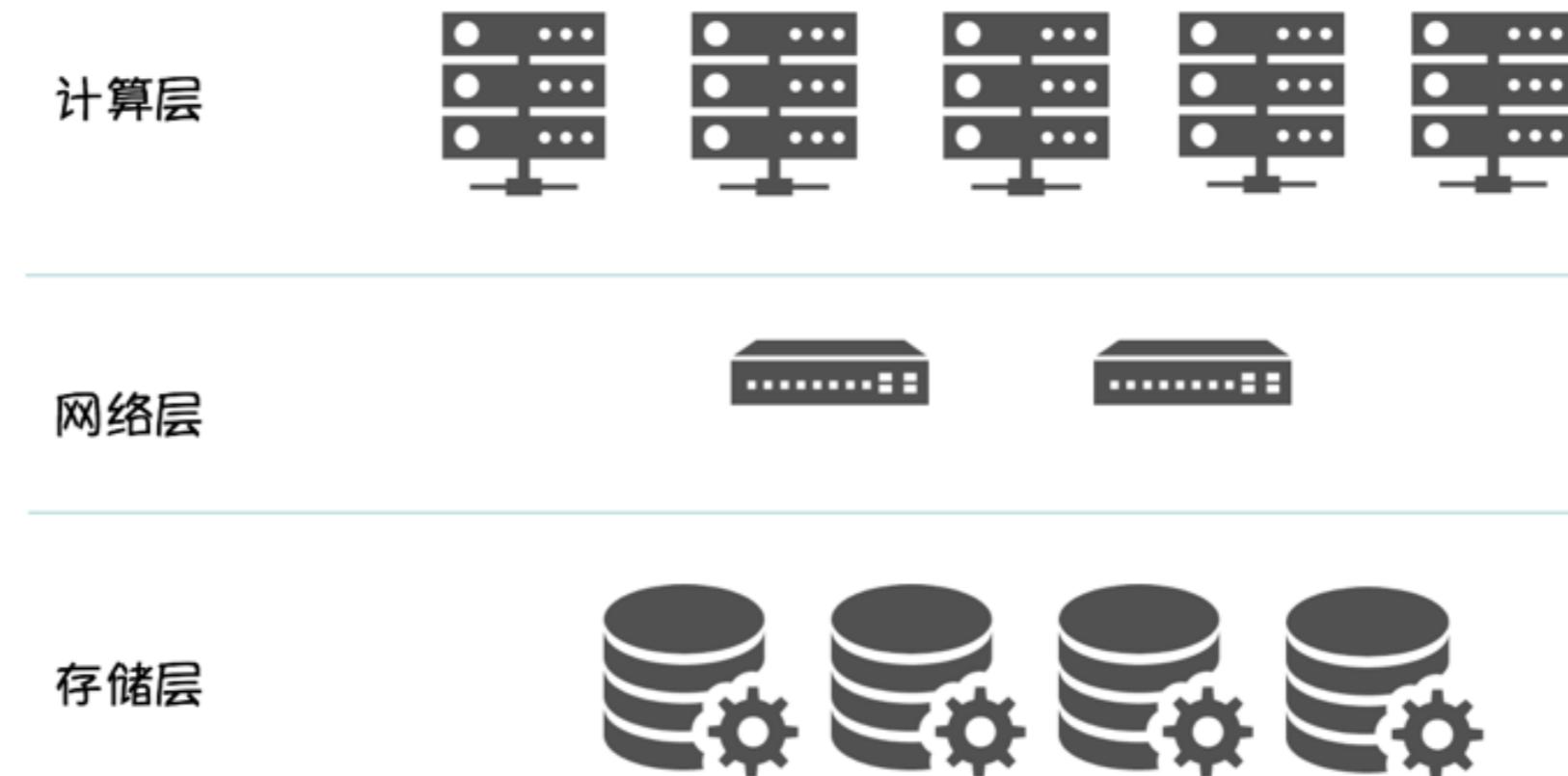


基于大数据分布式存储系统 Alluxio的负载均衡优化

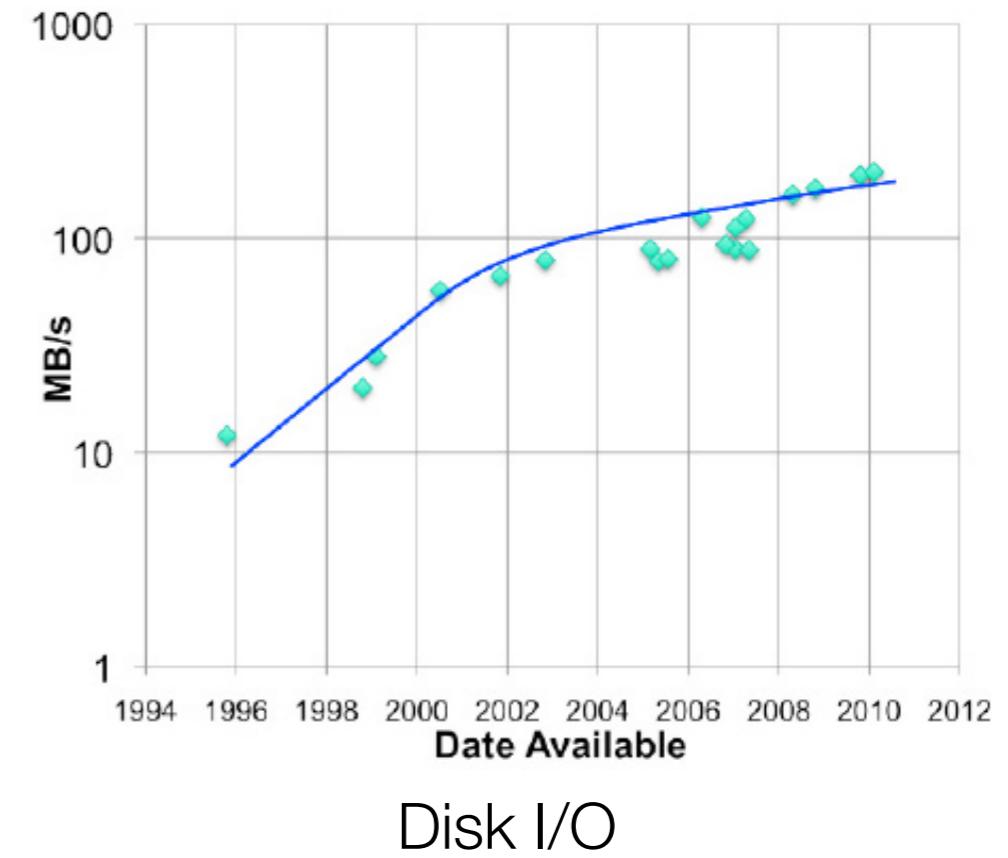
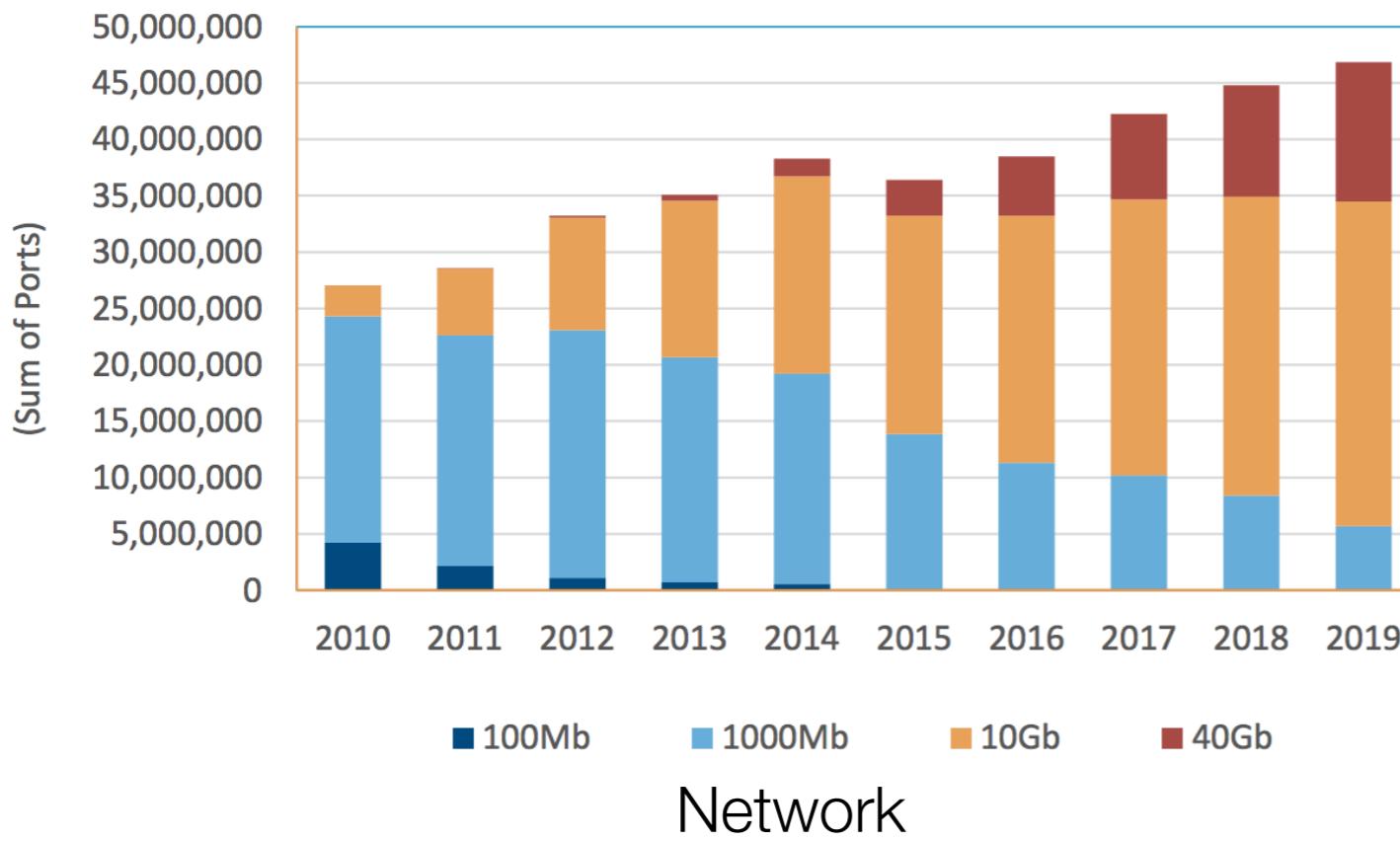
存储与计算分离

- 弹性部署
- 细粒度容量伸缩
- 快速扩容



网络和硬盘I/O速度

- 网络大幅提速
- 硬盘I/O速度的增长趋于**停滞**

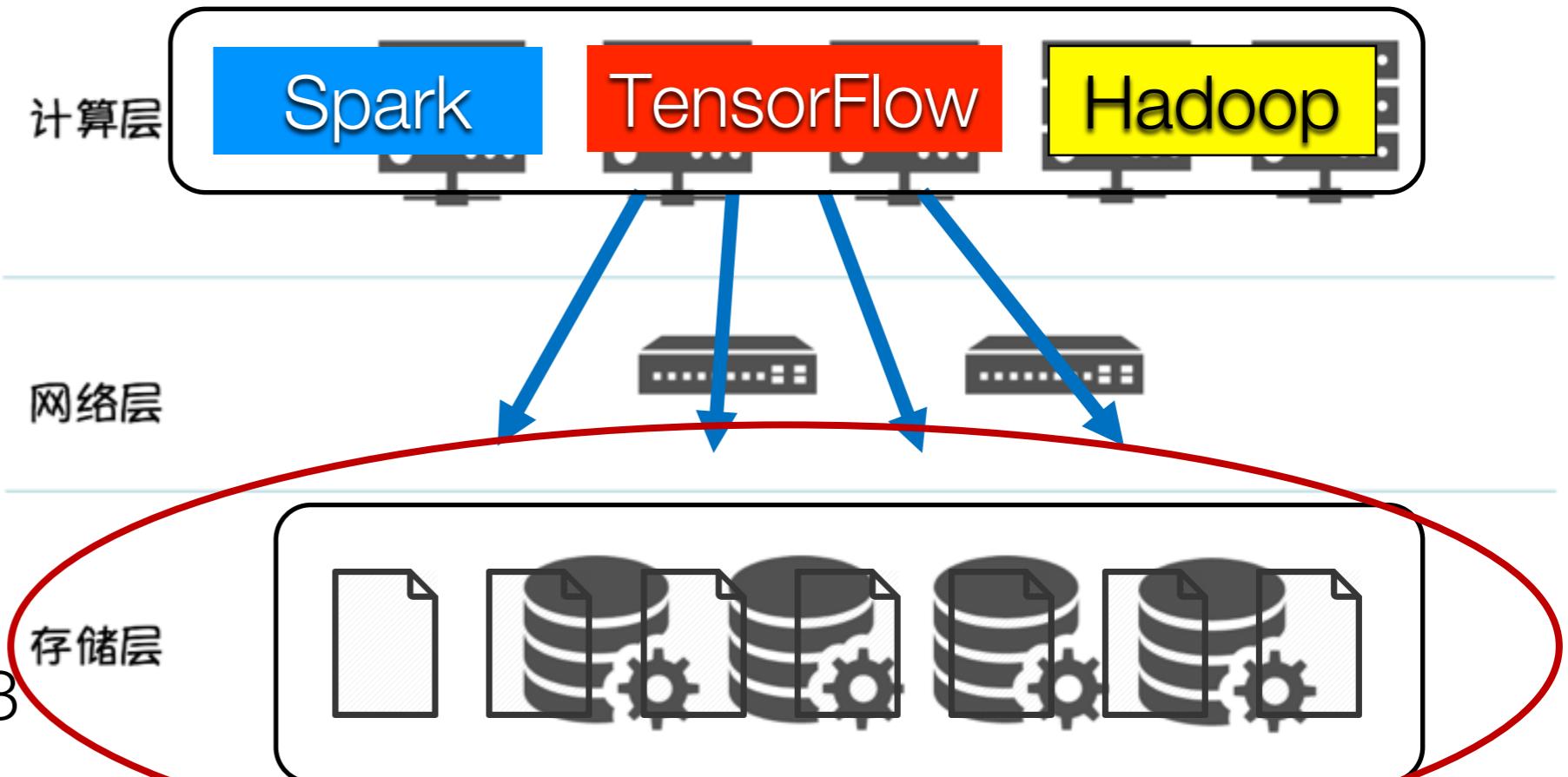


"The New Need for Speed in the Datacenter Network", IDC Technology Spotlight, 2015

云对象存储

- 硬盘本地性不再重要 [AMPLab, 2011]，得益于高速网络
- 硬盘I/O 仍然是瓶颈

Big data apps



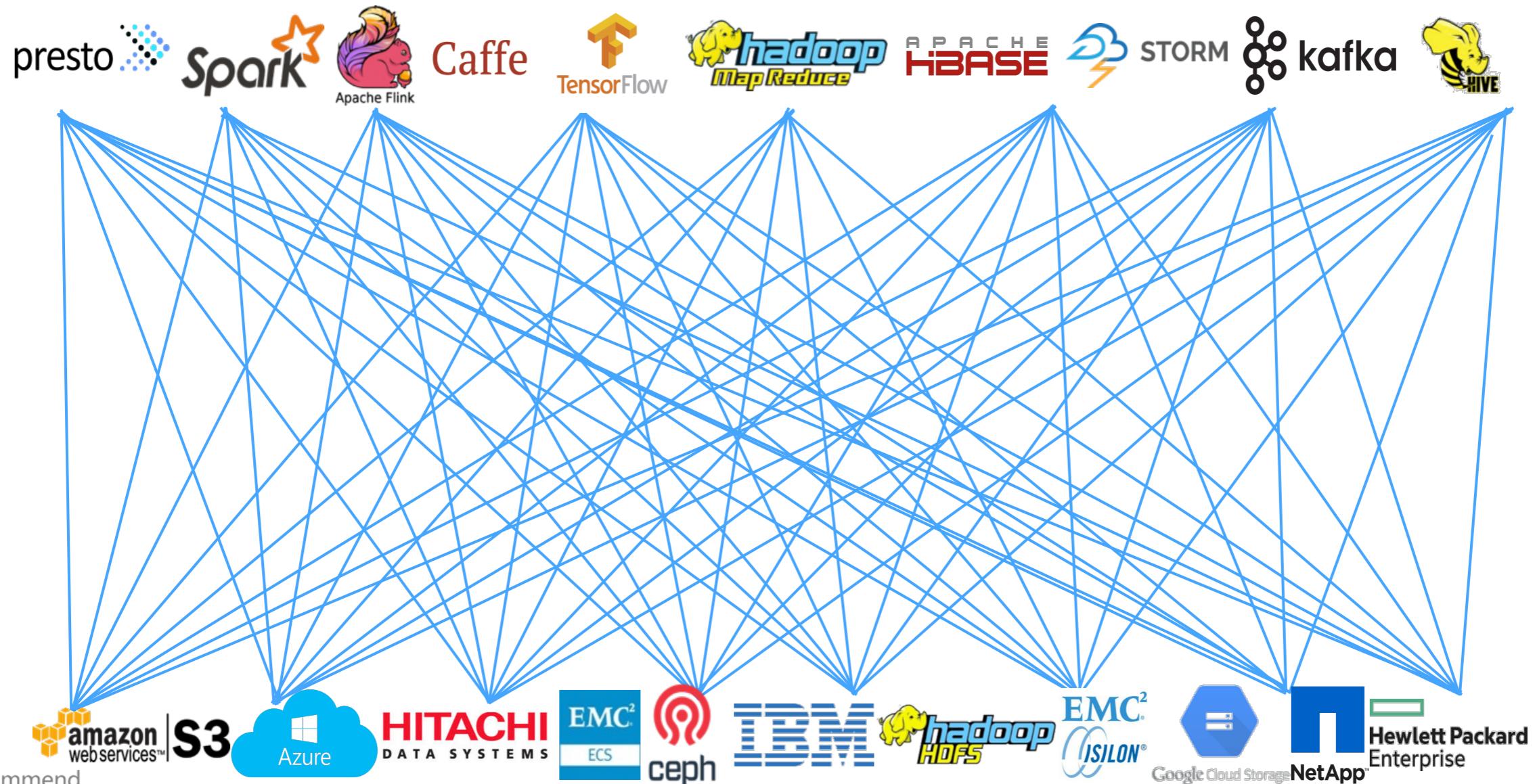
Cluster storage

HDFS, Azure Storage, S3

“Disk-Locality in Datacenter Computing Considered Irrelevant”, AMPLab, 2011

Data Ecosystem 1.0

- 多种计算平台、多种云存储
- 数据管理复杂度高、性能差、运维成本高

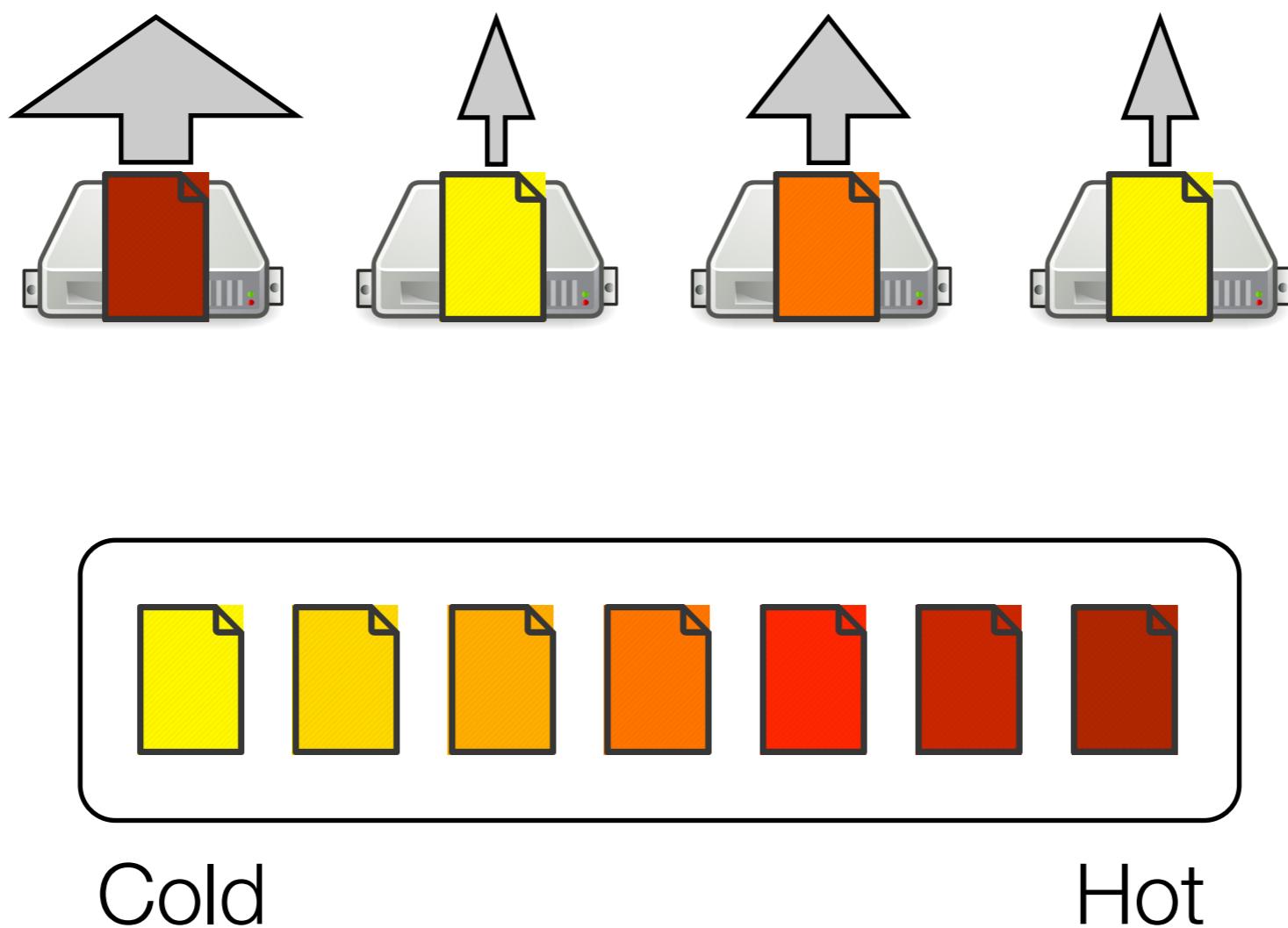


Alluxio: 统一化分布式内存文件系统

Data Ecosystem 2.0

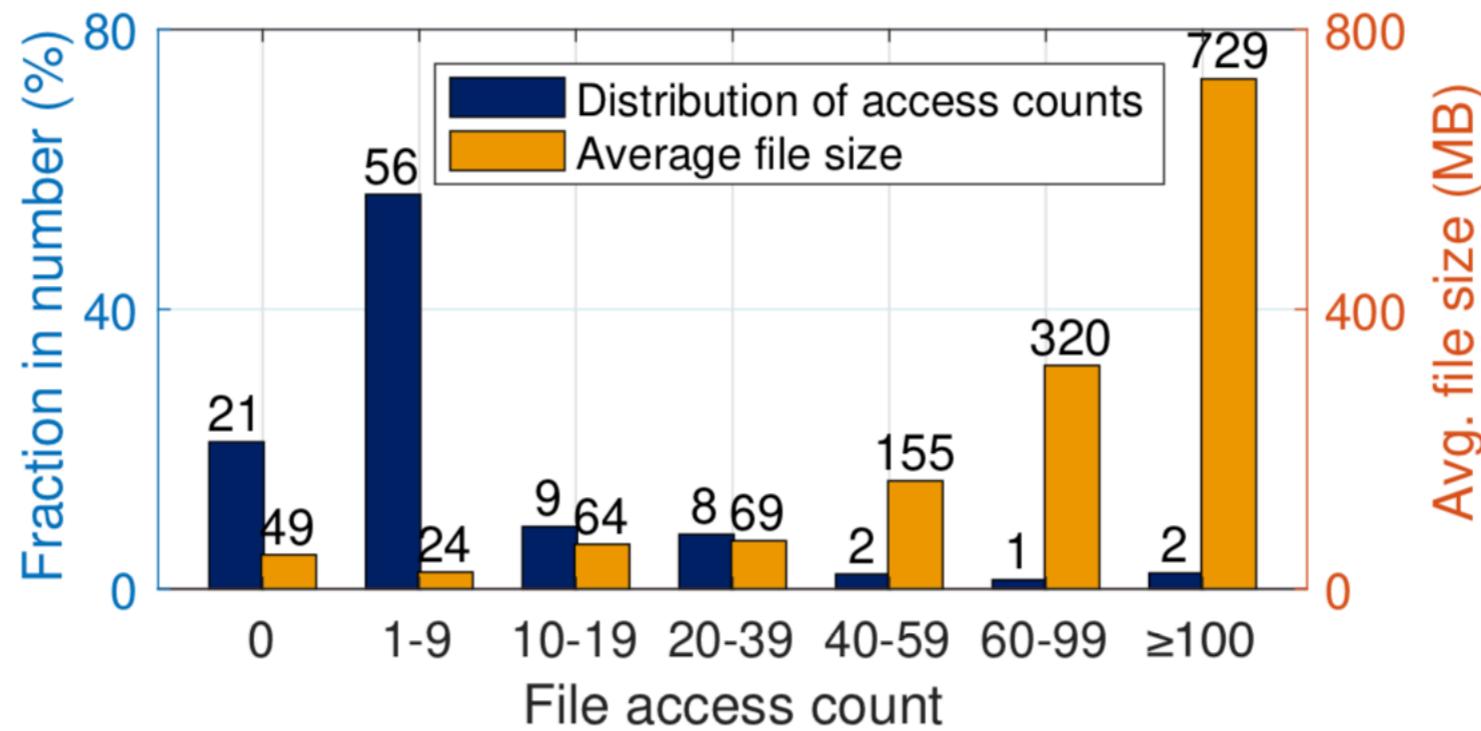


负载失衡：分布式内存系统 面临的巨大挑战



热度差异

Trace study in a Yahoo! cluster



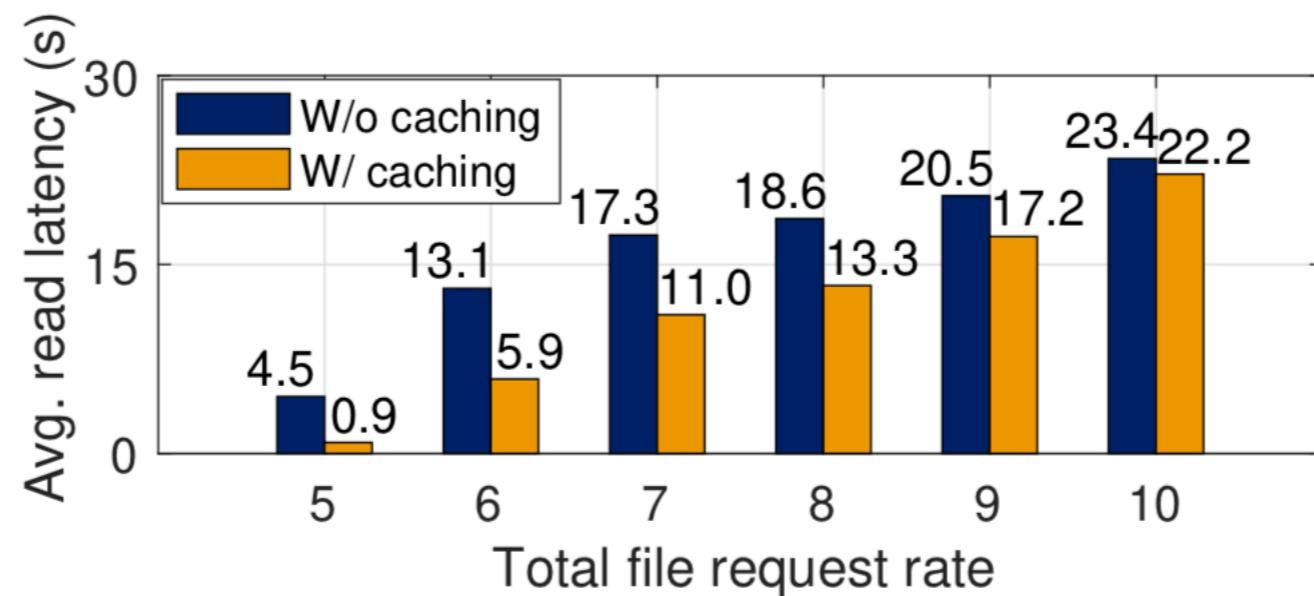
少数文件被高频率访问 ([Zipf-like 分布](#))

热点文件往往是大文件

负载失衡抵消了内存优势！

AWS EC2上一个30节点的Alluxio 集群

文件热度服从Zipf分布



Benefits of caching diminishes!

现有的负载均衡算法

State-of-the-art

Selective replication

[选择性复制]

Scarlett [Eurosys'11]

[Hong et al. SoCC'13]

Erasure coding

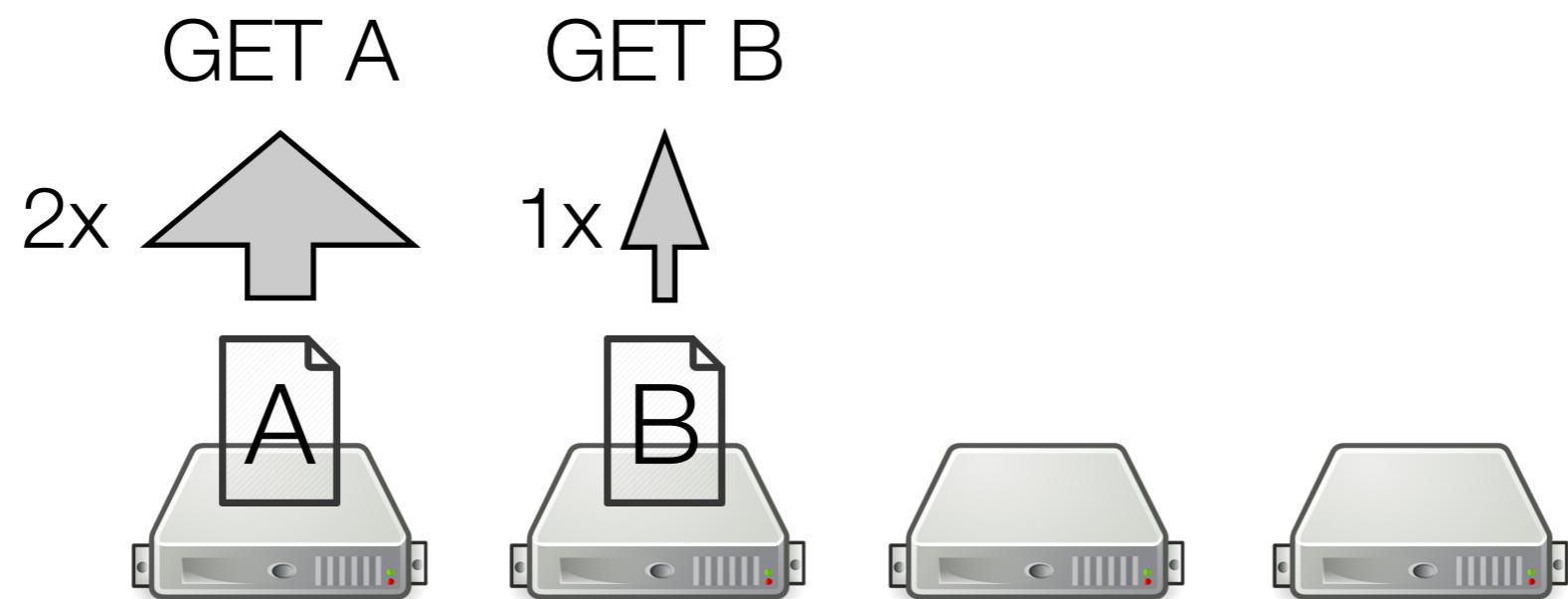
[纠删码]

EC-Cache [OSDI'16]

选择性复制

基于文件热度进行选择性复制

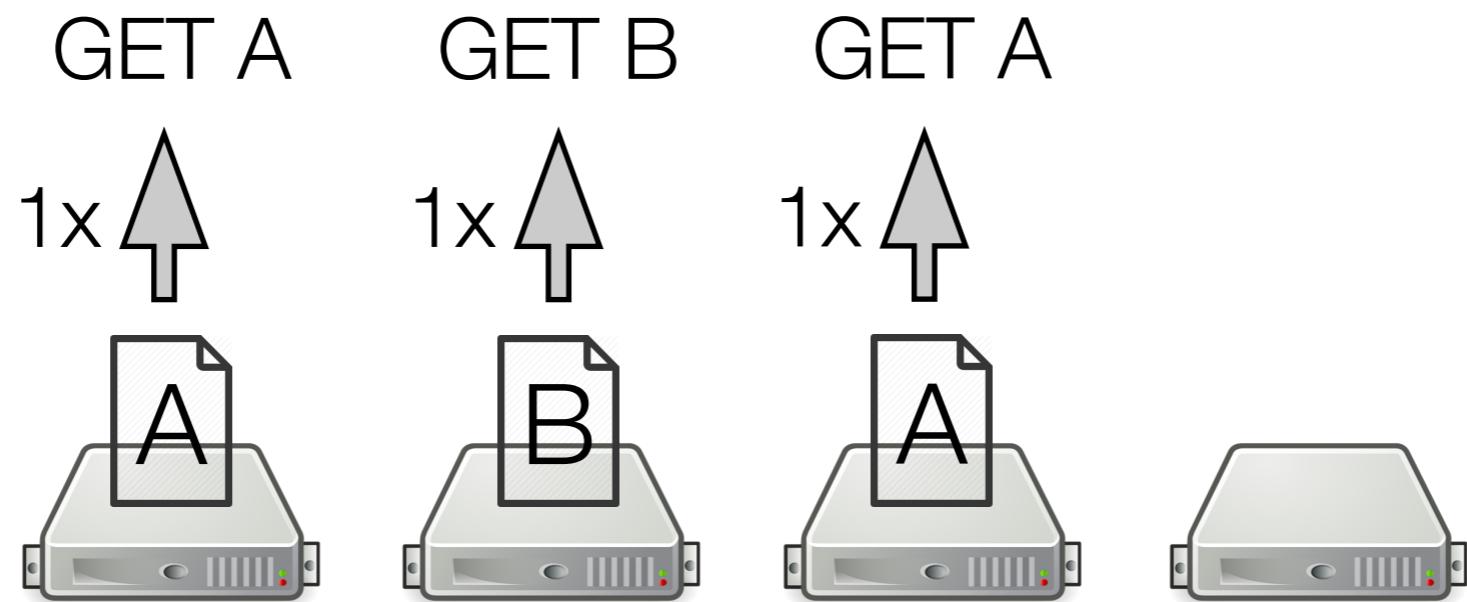
- 越热的文件被复制越多份



选择性复制

基于文件热度进行选择性复制

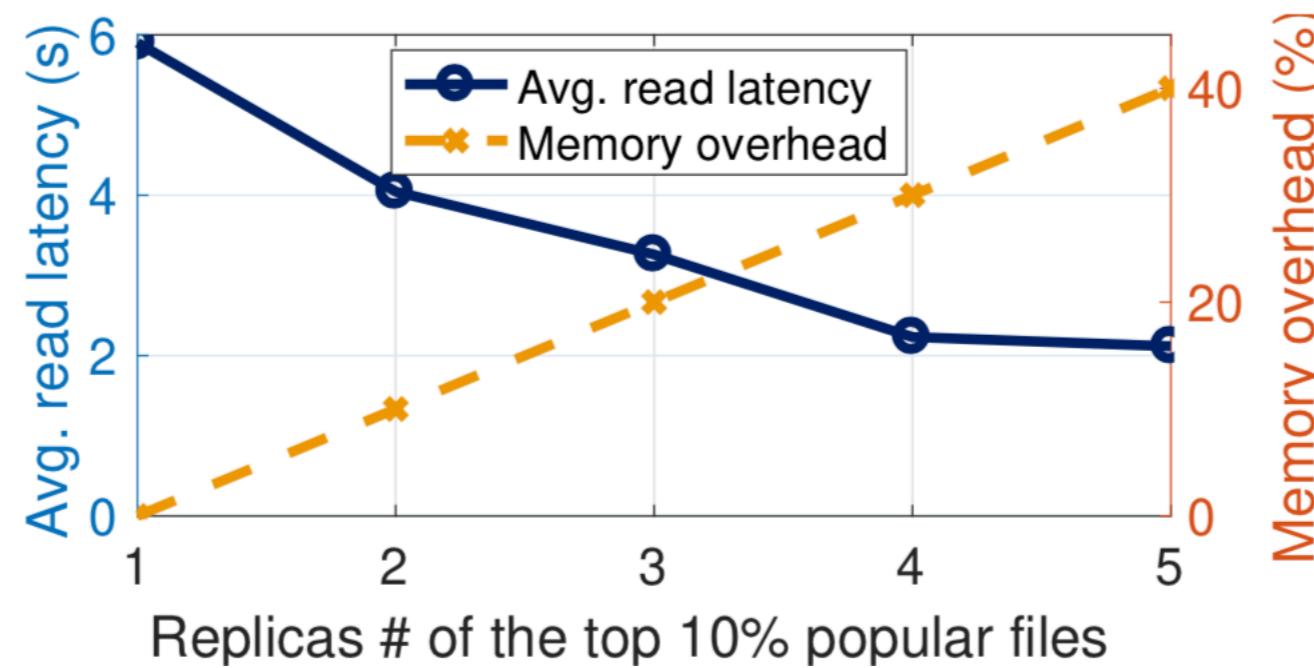
- 越热的文件被复制越多份



内存冗余

复制造成大量冗余

- 每一份复制增加**一倍**冗余
- 热点文件是大文件



纠删码

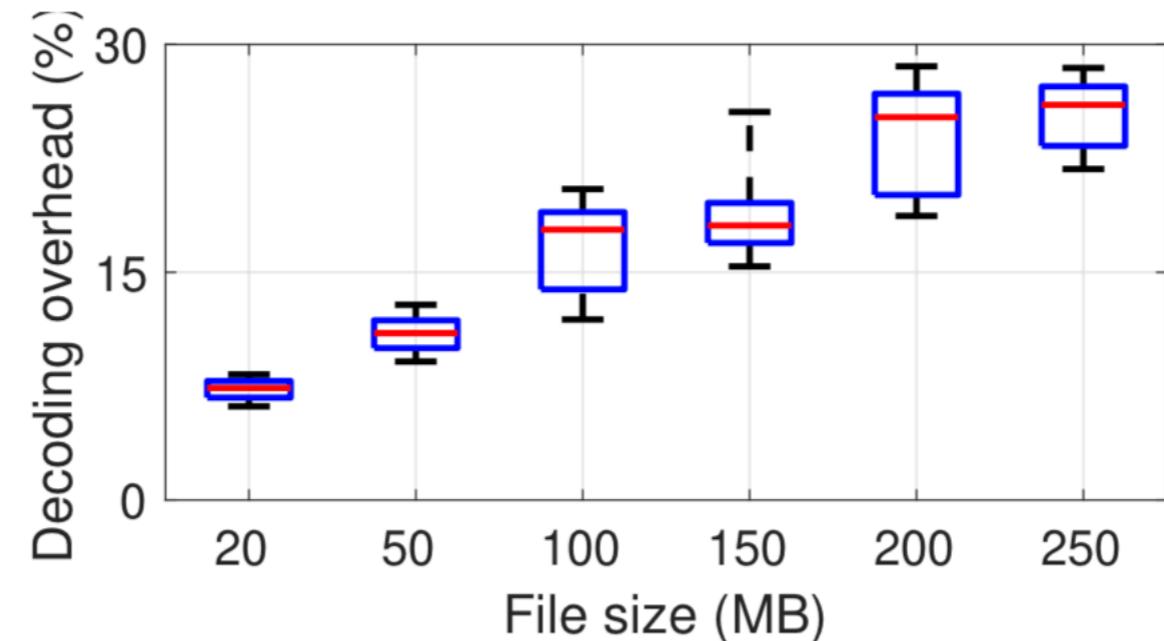
(k, n) 纠删码: 先将文件分为 k 个小块 (信息块) , 基于此编码生成另外 $n-k$ 个校验块

任意 k 个小块可以解码出原文件

➤ 更小的冗余度 : $(n-k)/k < 1$

编解码计算开销

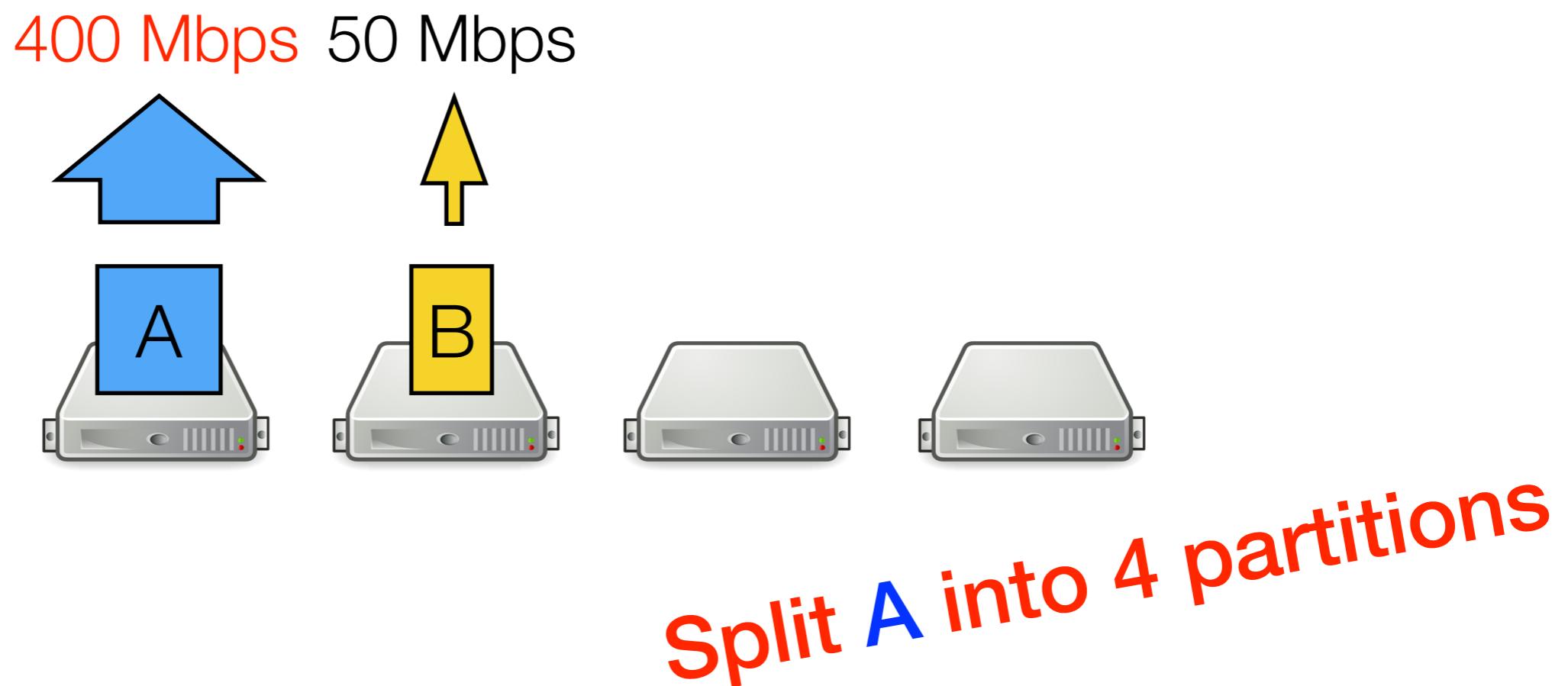
解码时间占整个读延迟的比例高达30%



我们的解决方案：选择性文件分割

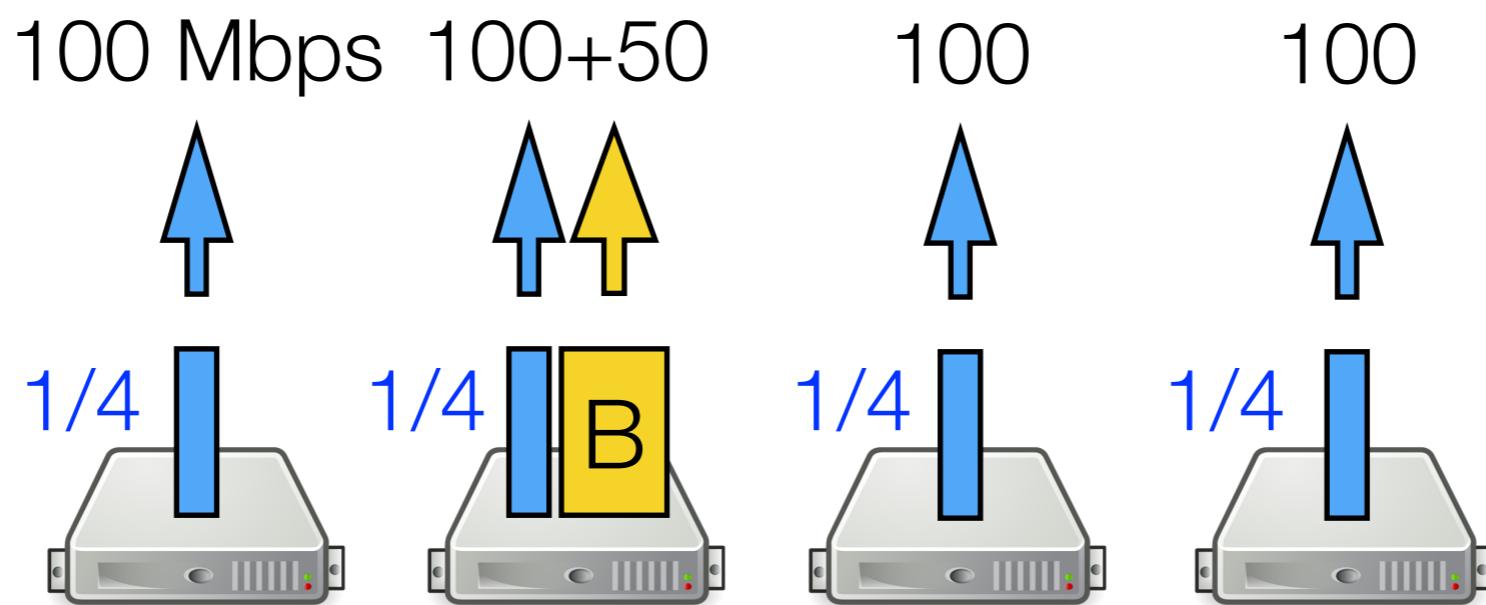
文件分割

将文件分割为多个小块，随机分散在内存系统中



文件分割

将文件分割为多个小块，随机分散在内存系统中

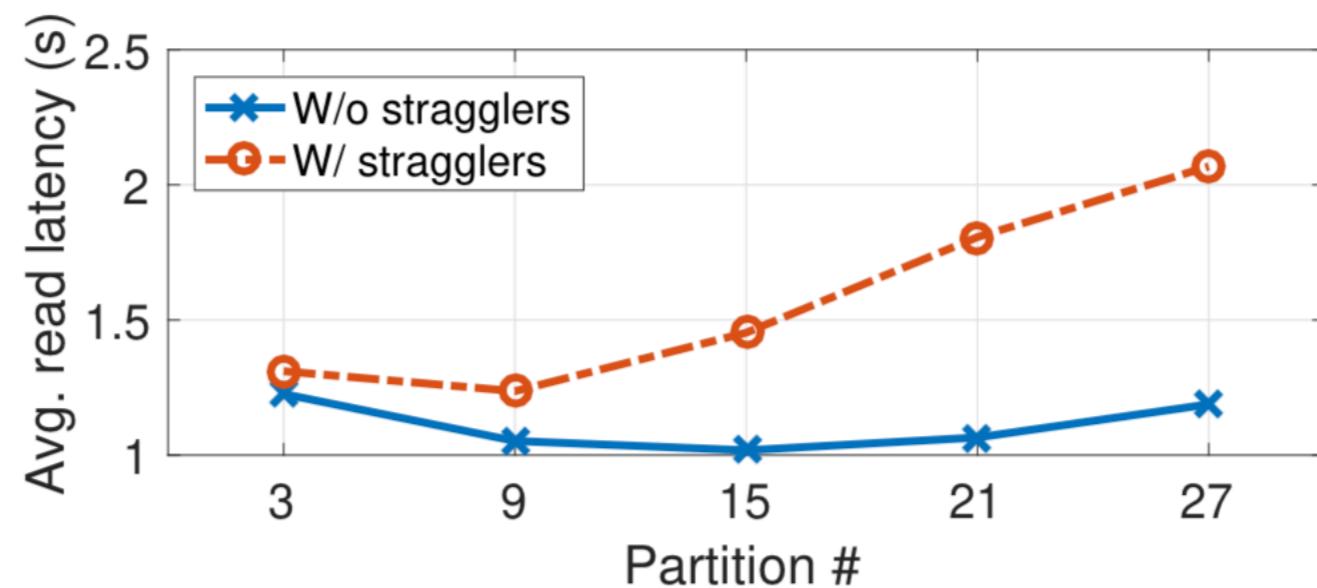


Split A into 4 partitions

文件分割的好处

- ✓ 将热点文件的负载分散在多台机器上
- ✓ 零冗余
- ✓ 无需编解码计算
- ✓ 并行I/O减少读延迟

然而,过多的文件分割数目会增加
straggler 节点 (拖后腿者) 的隐患



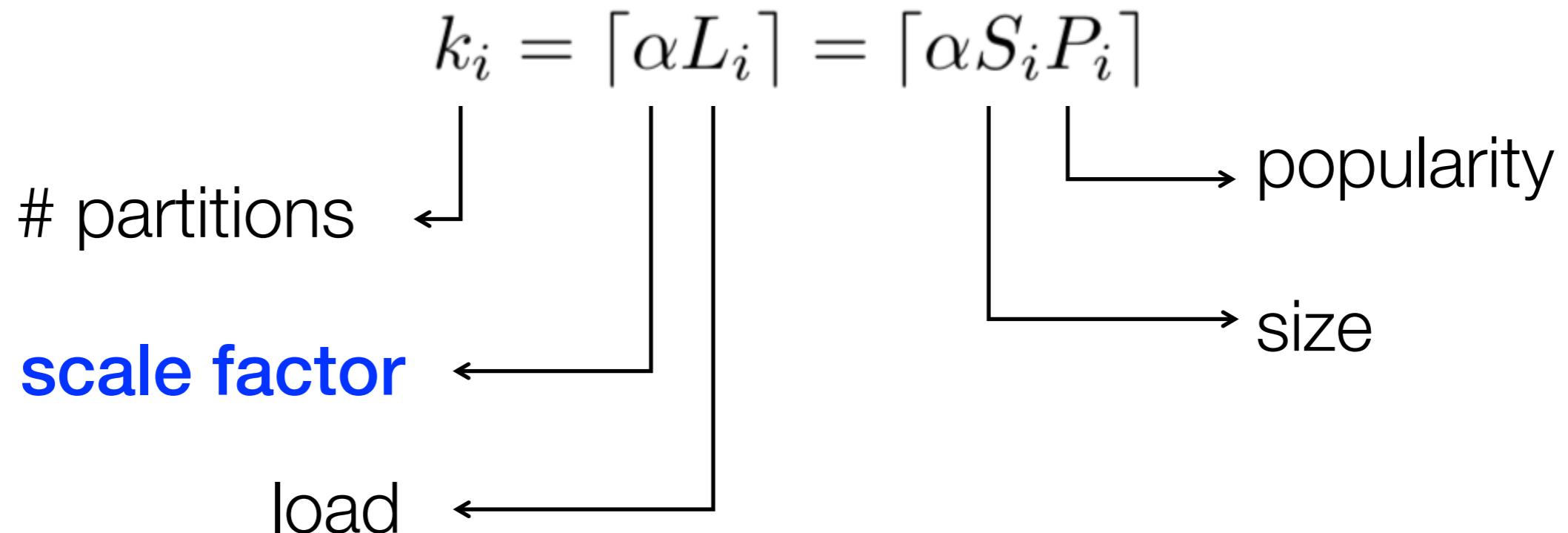
30-node Alluxio cluster, w/ and w/o stragglers

如何决定文件的分割数目？

- 太少了不足以均衡负载
- 太多了受straggler的影响

选择性分割

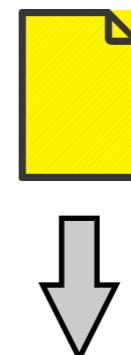
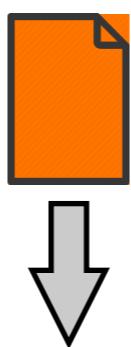
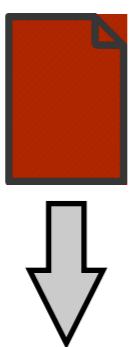
SP-Cache: Selective Partition



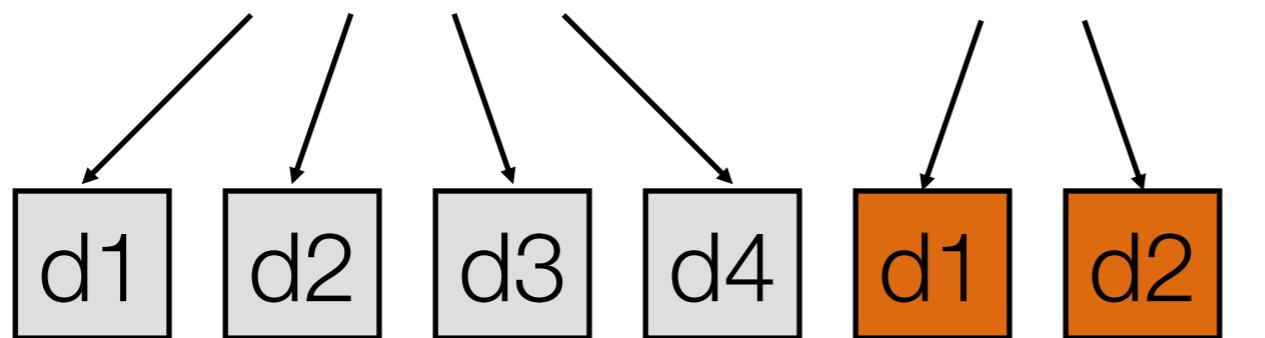
400 Mbps

200 Mbps

100 Mbps



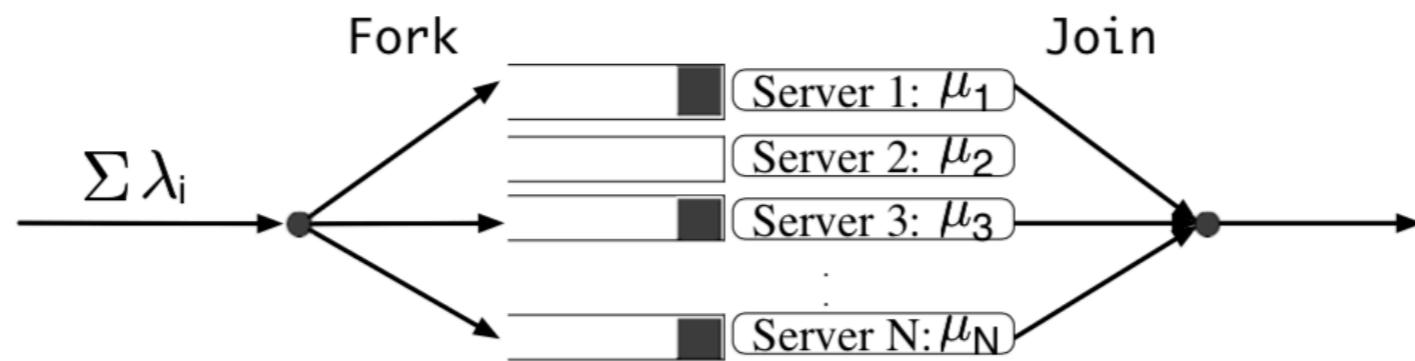
Selective partition



最优的Scale Factor

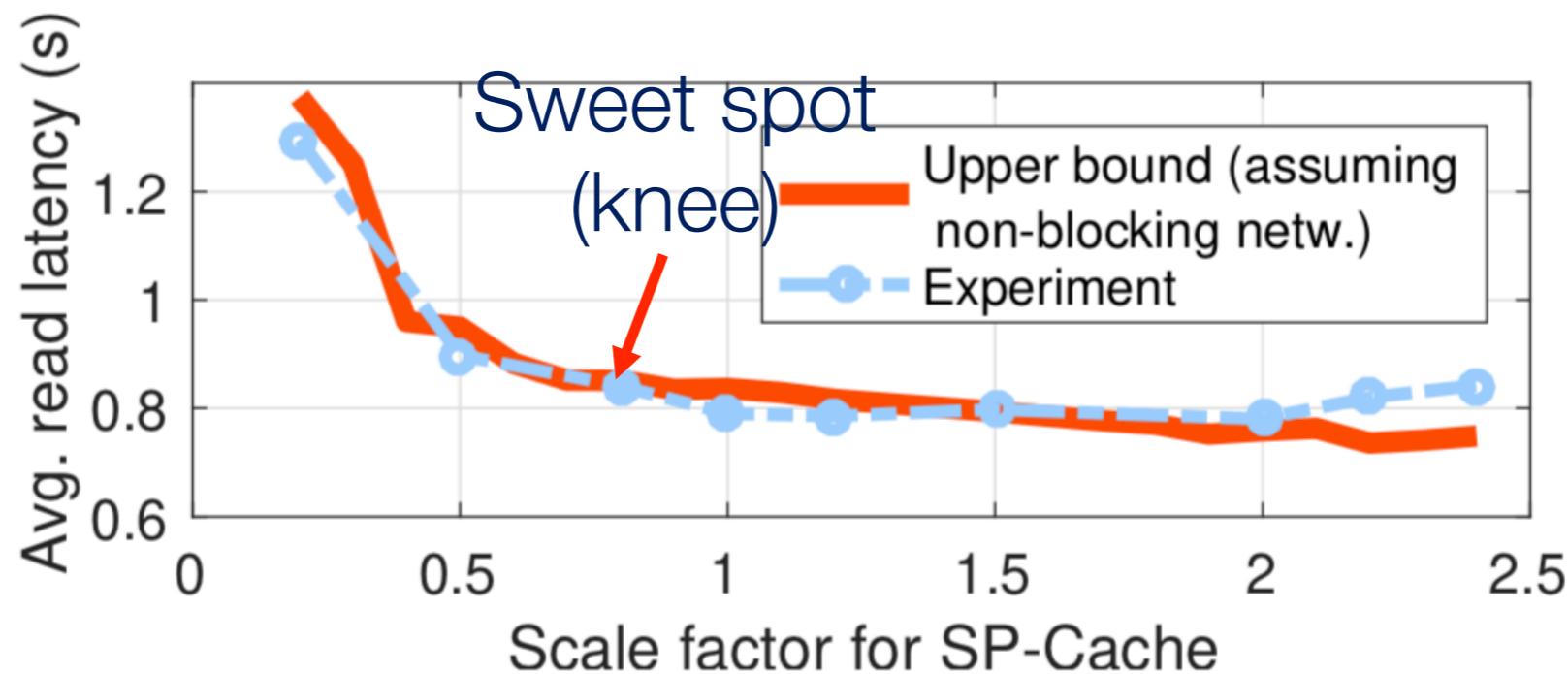
将分布式内存系统建模为一个fork-join queue

在没有straggler的假设下，分析平均读延迟



最优的Scale Factor

平均读延迟的理论上界



恰好足够均衡负载的文件分割数目

性能评价

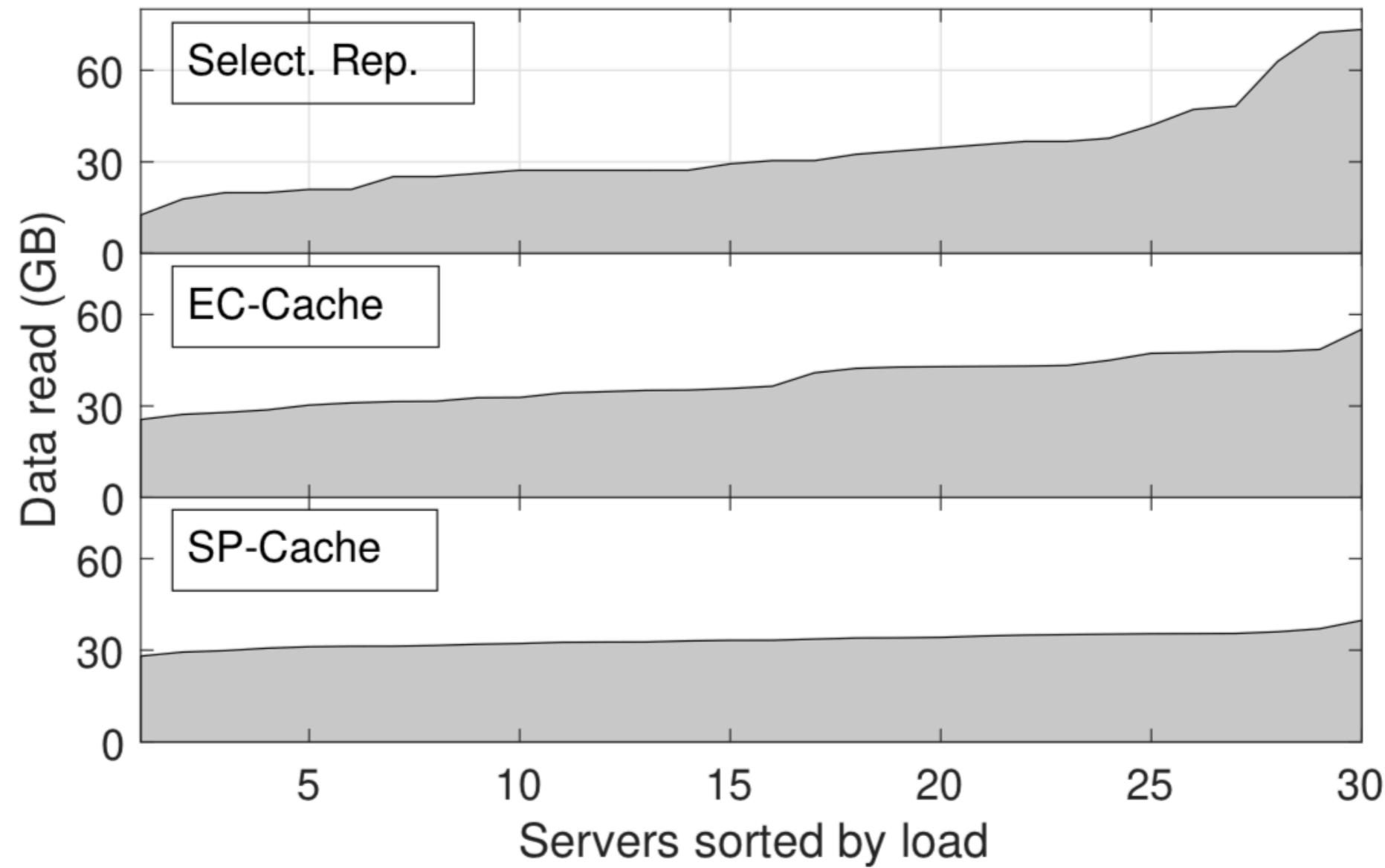
51个 r3.2xlarge 节点: 1 master, 30 servers 和 20 clients

文件热度服从Zipf分布

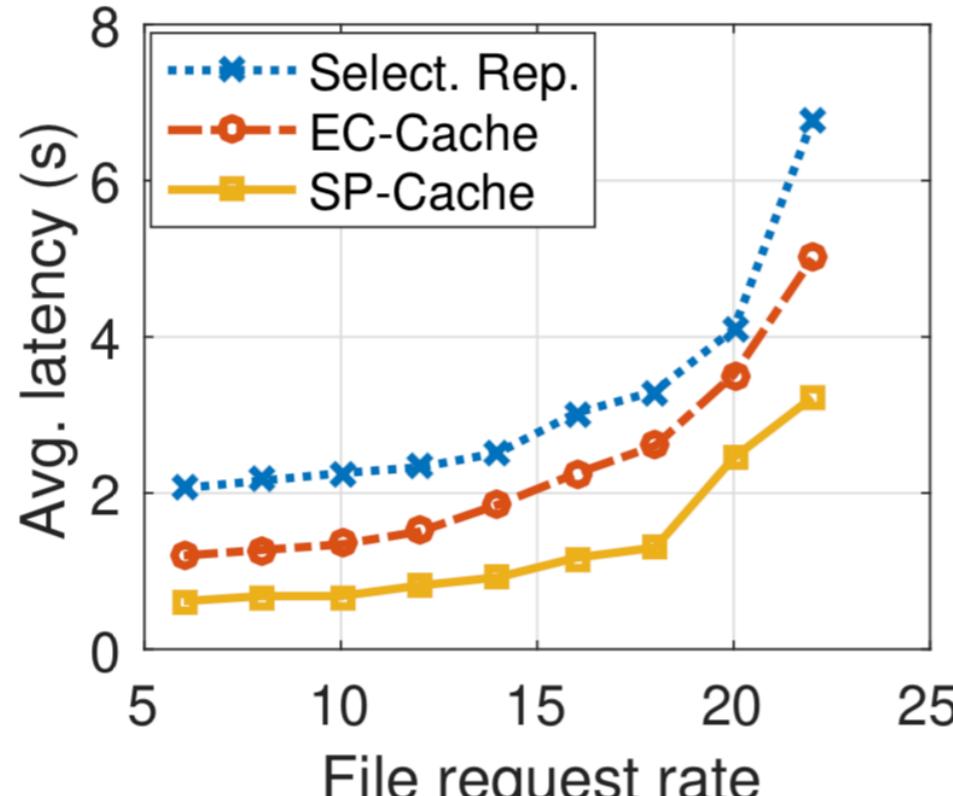
两个的现有策略

- EC-Cache: (10, 14) 纠删码 [Rashmi et al. OSDI'16]
- Selective replication: 将top 10% 的热点文件各复制四份

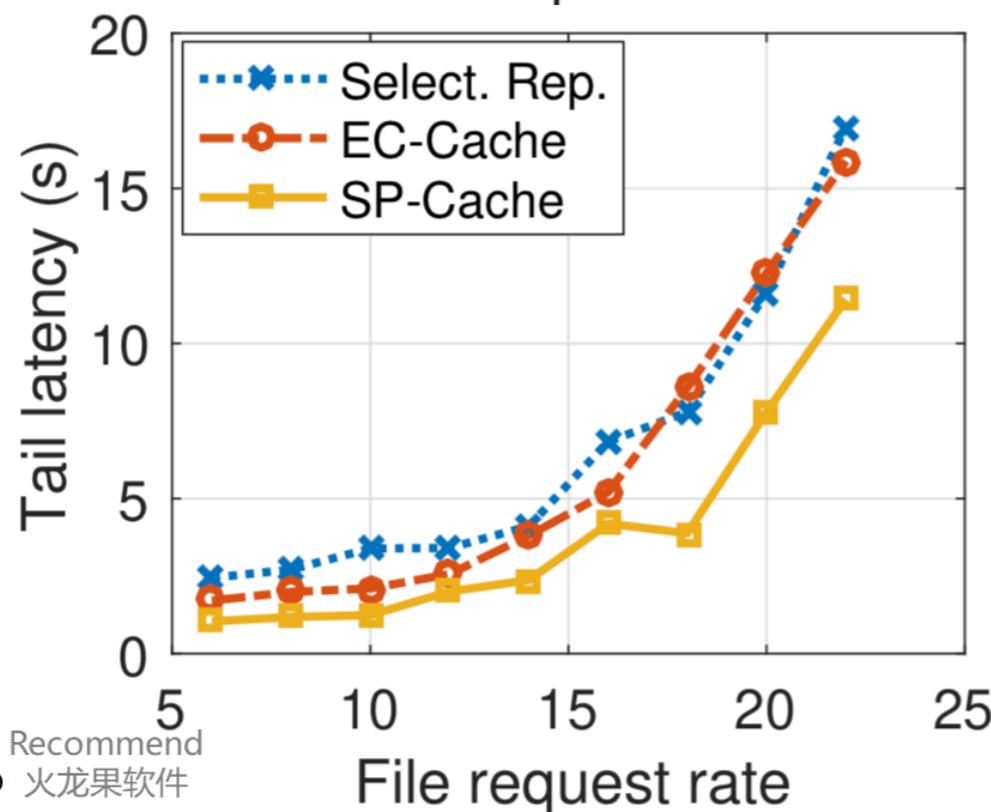
负载均衡



读延迟

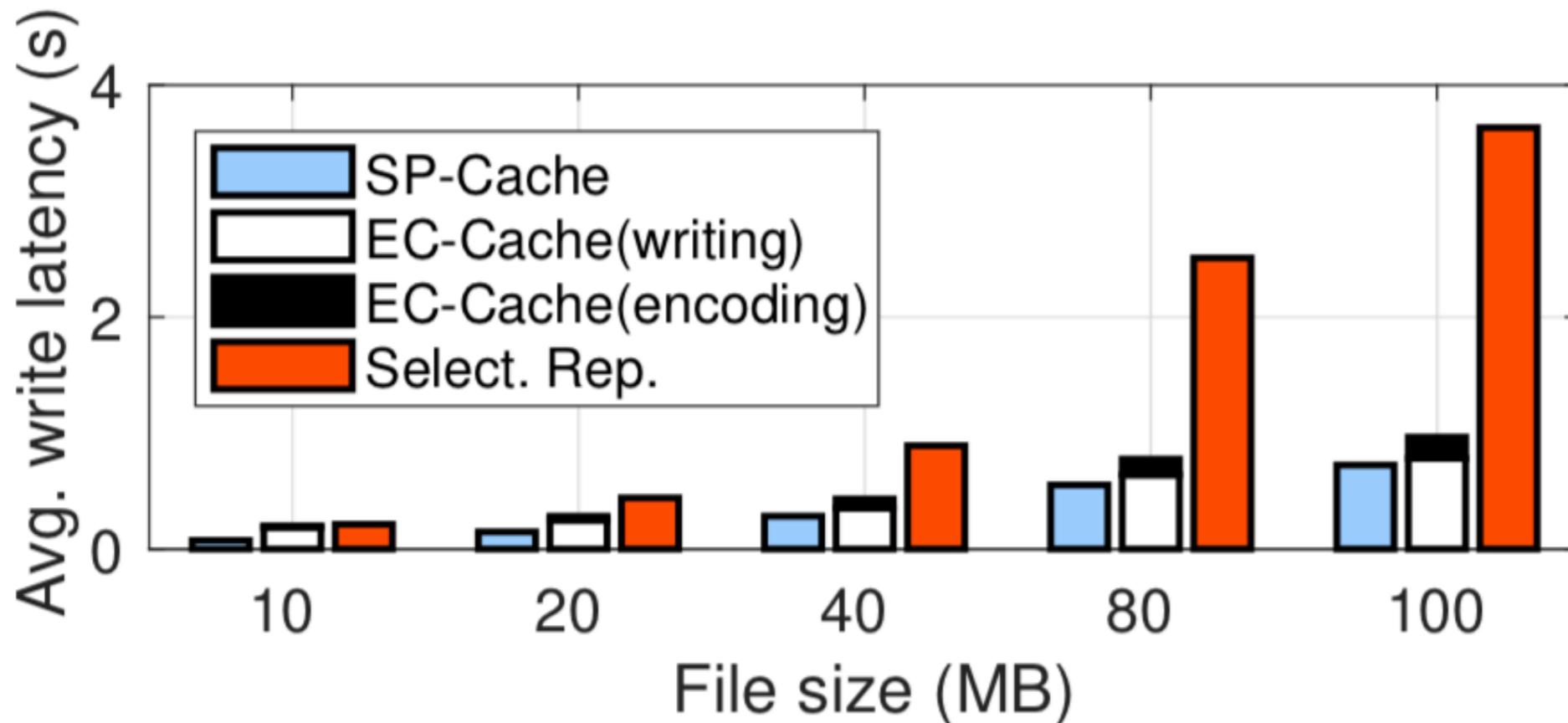


29-50% improvement
over EC-Cache



22-55% improvement
over EC-Cache

写延迟



1.77x and 3.71x faster than EC-Cache and selective replication **in average**, respectively

总结

- Alluxio: 统一化分布式内存文件系统
- 选择性热点数据分割，实现分布式内存系统负载均衡
 - ▶ 恰好足够的分割数量
 - ▶ 零冗余
 - ▶ 零编解码开销

谢谢关注！
