Semi-hierarchical Semantic-aware Storage Architecture

半层次化的语义存储体系结构

华宇
华中科技大学

https://csyhua.github.io

当前存储系统的发展趋势

S M I L E

当前存储系统的发展趋势1

- SMILE
- Scale-规模化：大数据，大存储

当前存储系统的发展趋势2

- SMILE
- NN(M)-智能化：

当前存储系统的发展趋势3

- SMILE
- Integrated-一体化：

➢ Near Data Processing：

- ➢ Processing in-memory (PIM)
- ➢ In-storage computing (ISC)
- ➢ Quantx(美光), Optane(英特尔), NDP(华为), ……

当前存储系统的发展趋势4

- SMILE
- Long-term 长期化：

➢存储载体和运行环境

➢存储数据的时效性和价值

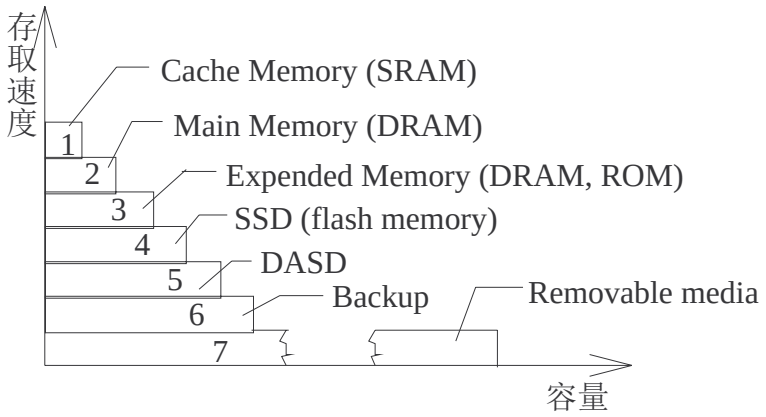
当前存储系统的发展趋势5

- SMILE
- Edge-边缘化：
- 边缘计算，雾计算，邻近计算, ……

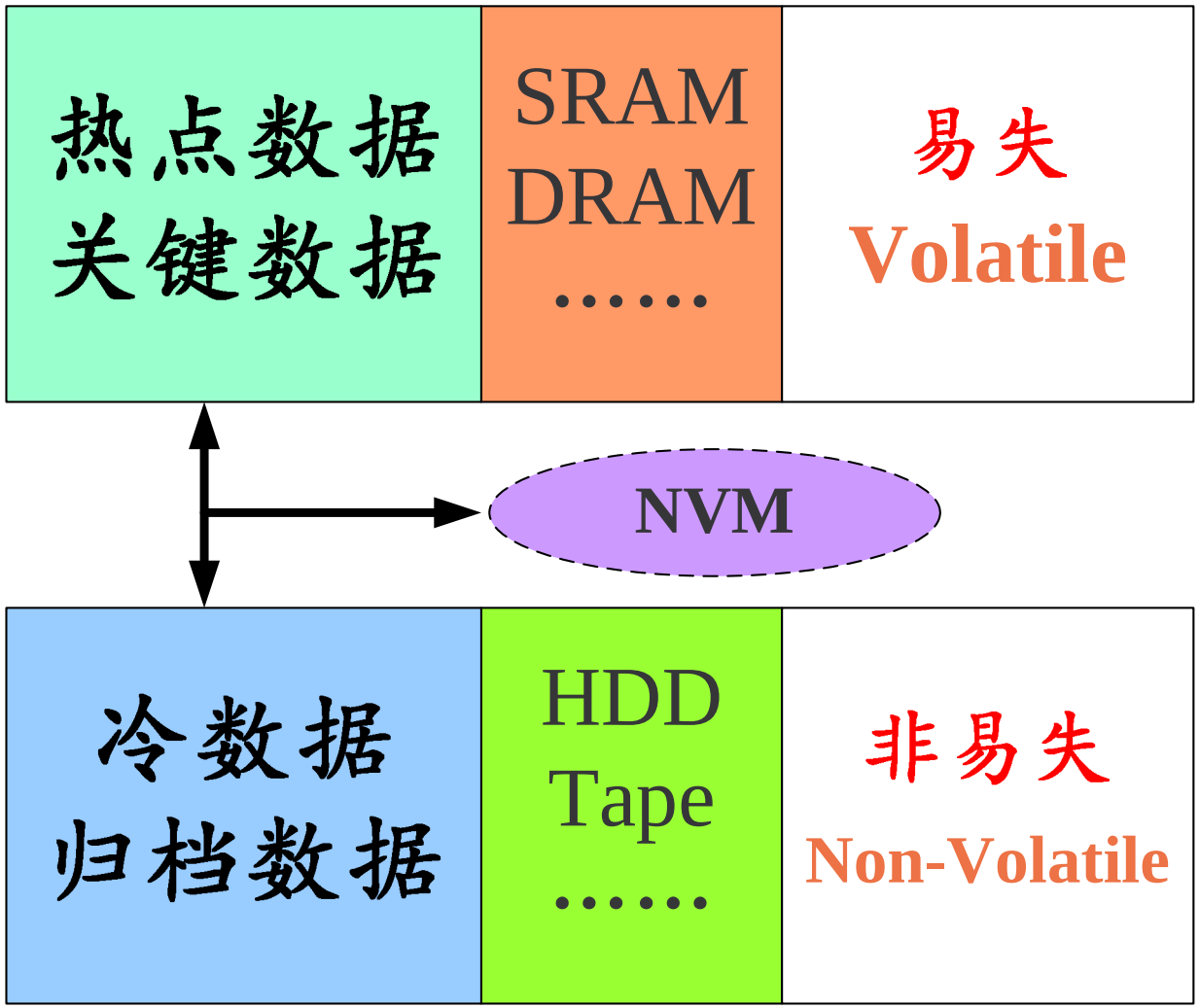
技术挑战：存储载体层次化

- 原理异构性
- 性能差异性
- 管理复杂性

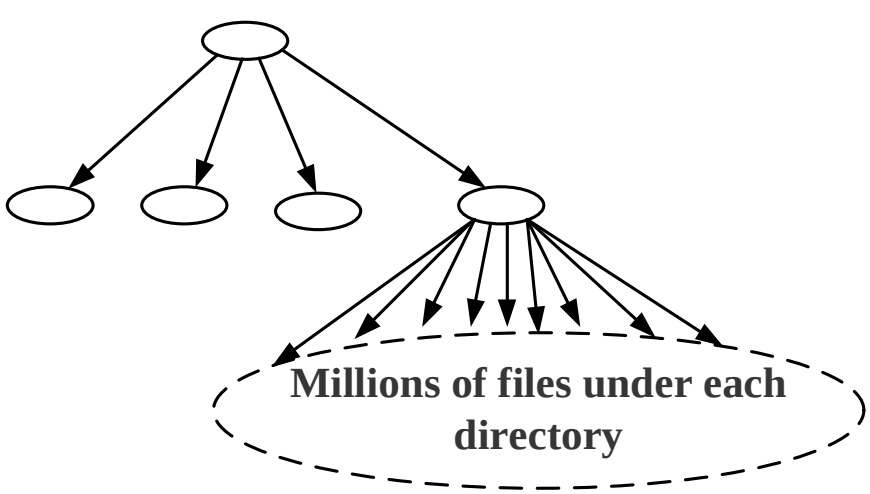
- 平房->金字塔->摩天大楼
- 层次越来越高

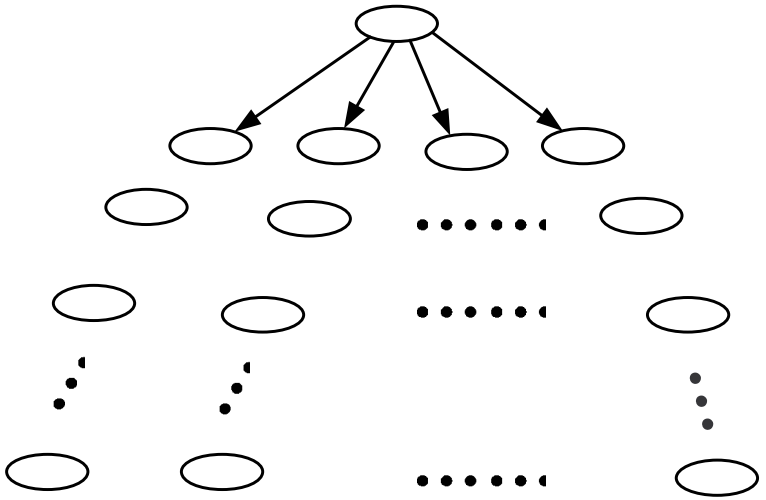


存储的安全和可靠：


热点数据
关键数据
SRAM
DRAM
……
易失
Volatile
NVM
冷数据
归档数据
HDD
Tape
……
非易失
Non-Volatile

层次化架构的组织模式





This tree is too FAT !

This tree is too HIGH !

- 顶层复杂化：由于规模和复杂度以及信息传递和迁移，导致少量中心节点管理全局数据变得困难，多节点设计面临可扩展性问题
- 边缘智能化：存储设备自身已有能力处理一定的操作。

垂直层次化架构

- **思路**：依托访问的局部性Locality，少量关键数据占用稀缺资源，即多等级的VIP策略
- 但是，目前Locality在变弱，而数据量剧增，使得提升hit ratio变得困难
- 信息传递更困难：跨层次，跨介质，占用有限通道
- **传输代价**：
  - 层次间总线
  - 节点间带宽
  - 数据中心间专线

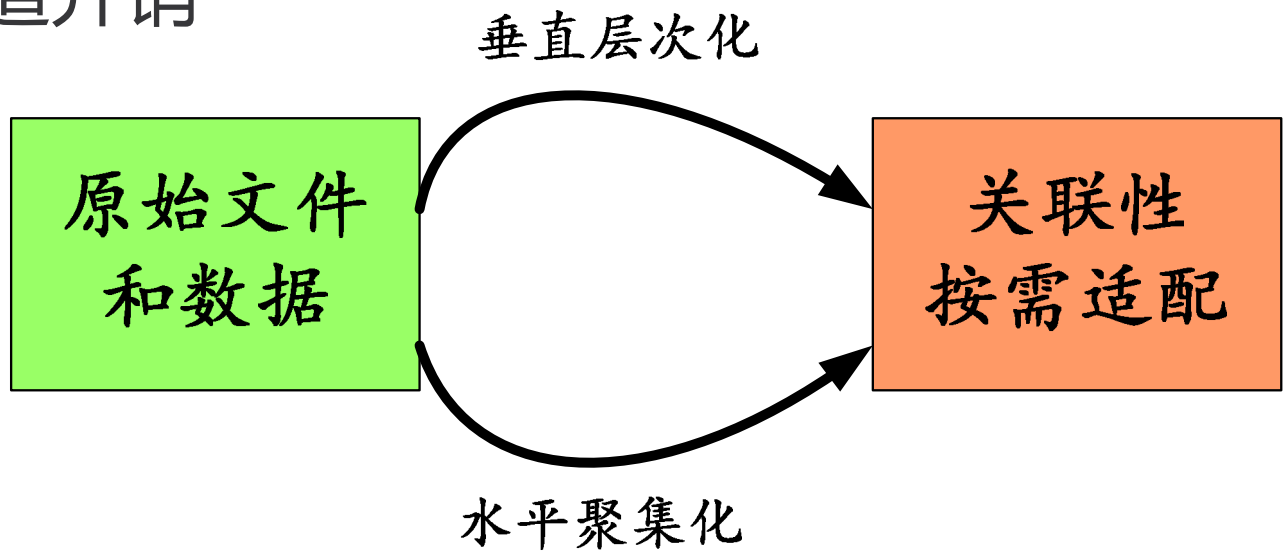
传统的层次化体系架构

- 机理：

遵从进化论，物竞天择，适者生存

层次化

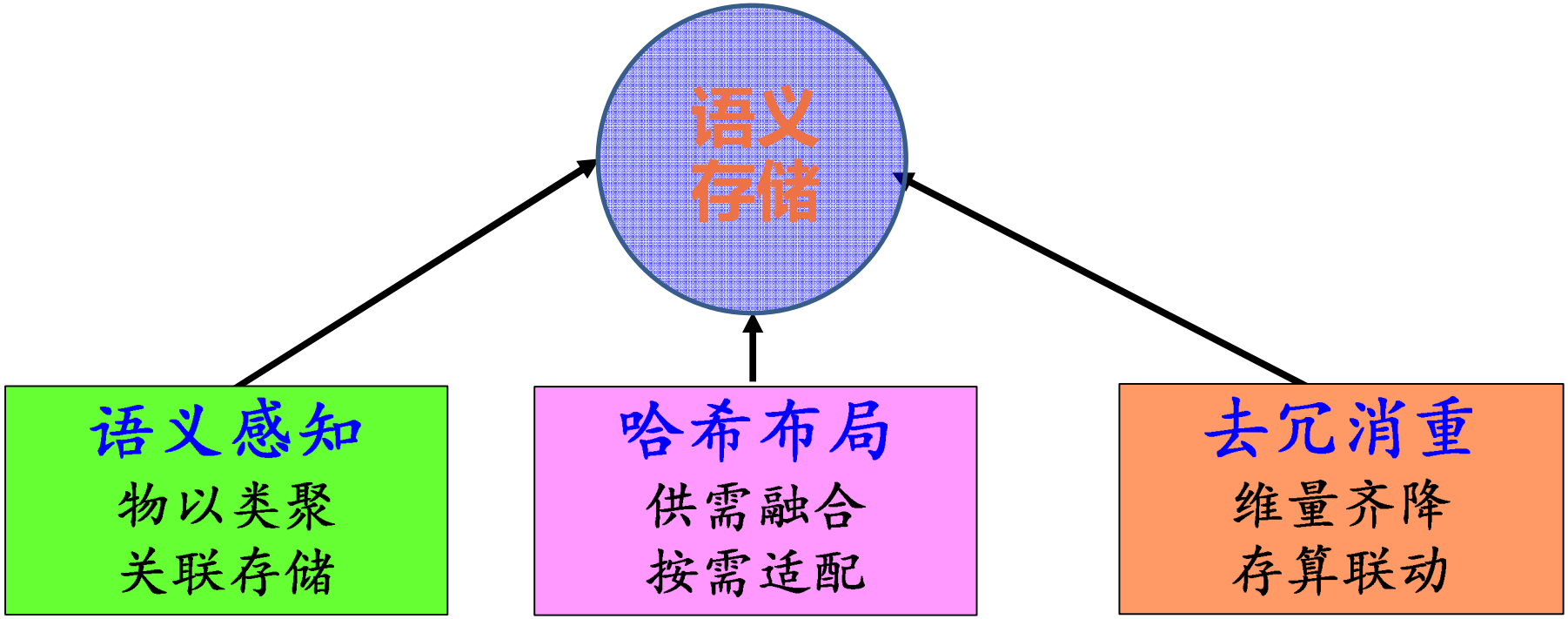
- **目的**：寻找关联性
- 层次化是一种动态筛数据的方式，其本质是要实现关联聚类，按需适配。
- 如果水平/扁平结构可以实现，也可以减少总线和通道开销



殊途同归

半层次化语义存储

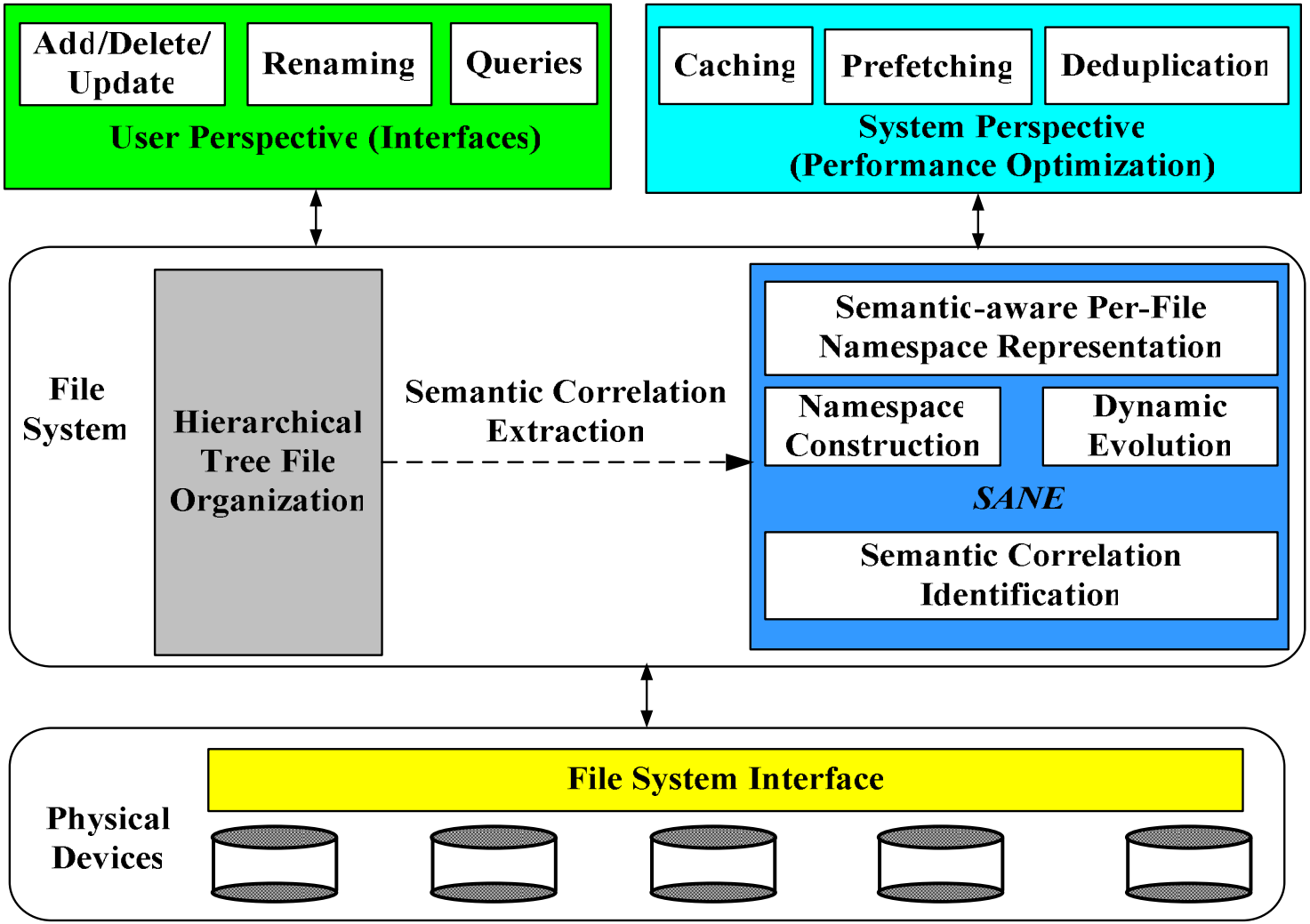
- 科学问题：
  - 如何在大规模存储系统中存储和组织海量数据
- 学术思路：
  - 语义存储作为存储系统的组织模式



主要相关工作

- **语义命名空间**：SANE(TPDS14)
- **语义数据聚集**：FAST(SC14), HAR(ATC14), SiLo(ATC11),
- **语义哈希计算**：SmartCuckoo(ATC17), DLSH(SoCC17), SmartEye(INFOCOM15), NEST(INFOCOM13)
- **语义在线应用**：ANTELOPE(TC14)

SANE: 系统体系结构
关联语义和数据实体

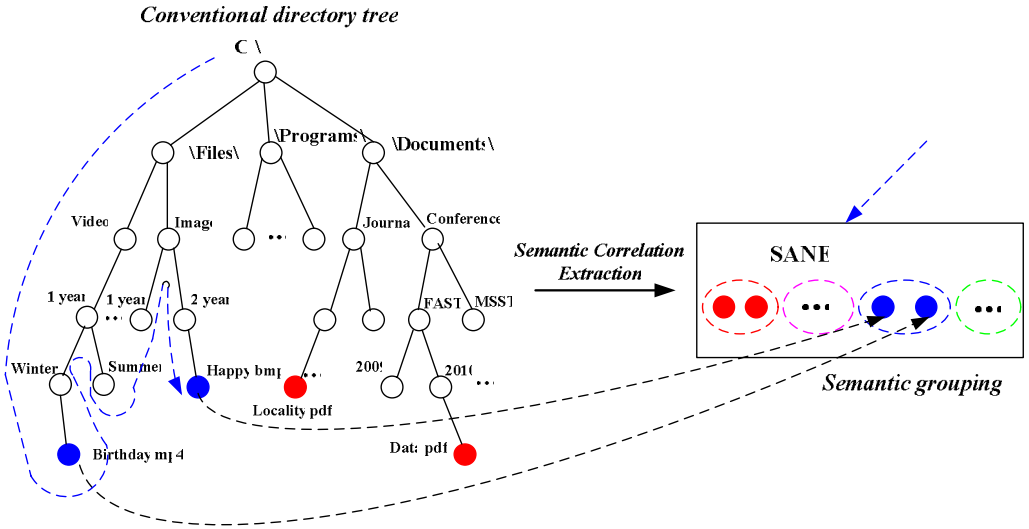


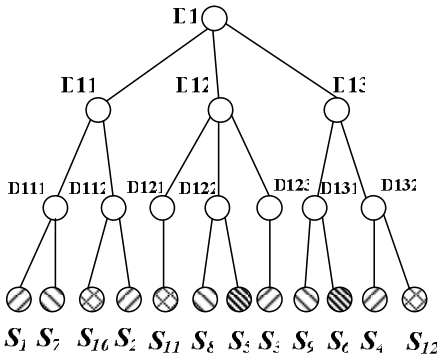
"SANE: Semantic-Aware Namespace in Ultra-large-scale File Systems", IEEE Transactions on Parallel and Distributed Systems (TPDS), Vol.25, No.5, May 2014, pages:1328-1338.

SANE: 大规模存储系统命名空间

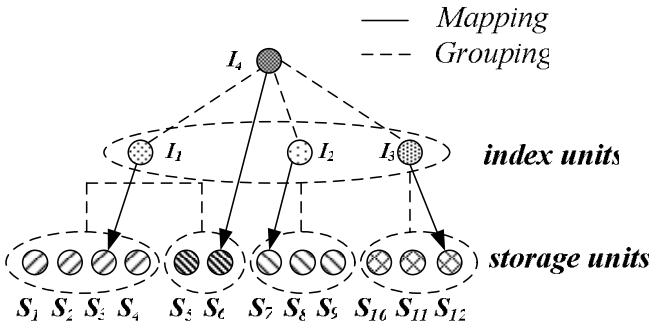
扁平化

- 海量存储系统的层次化文件结构是影响系统性能的主要瓶颈之一
- **设计的目的：**
  - 可检索
  - 唯一性





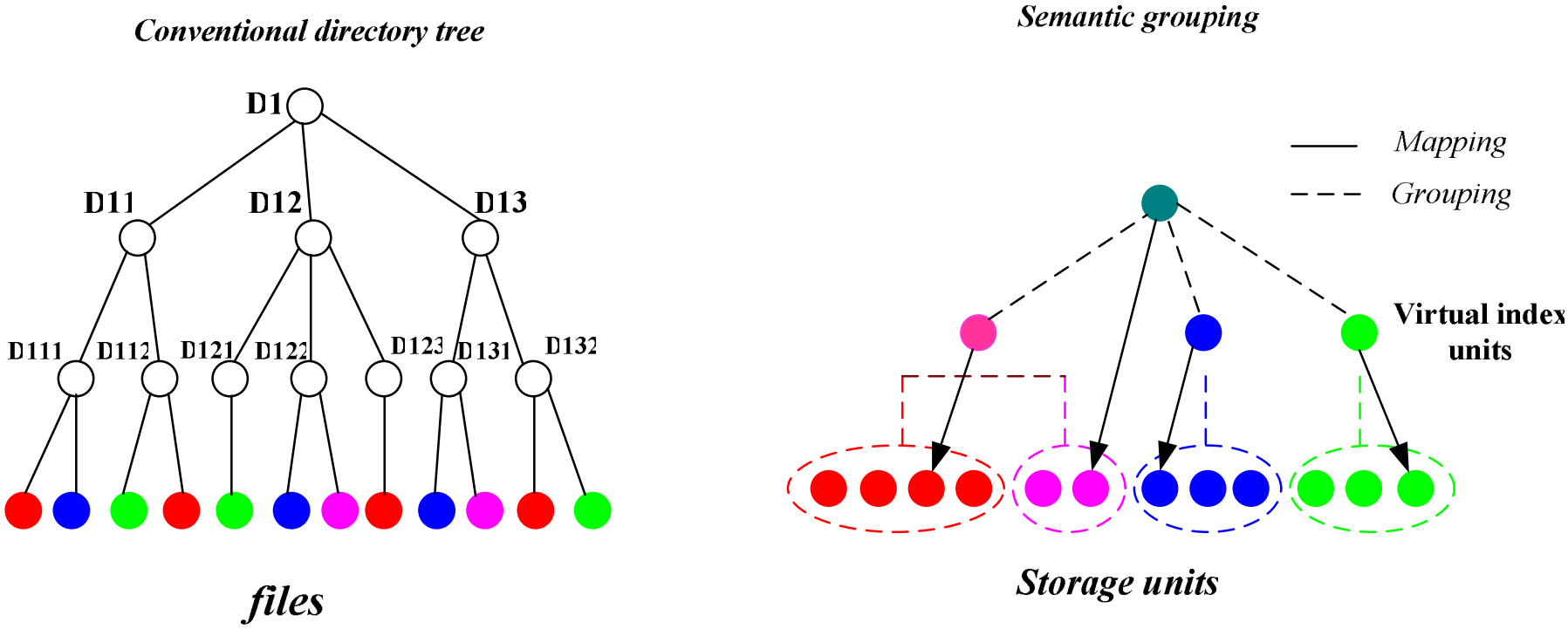
Conventional directory tree



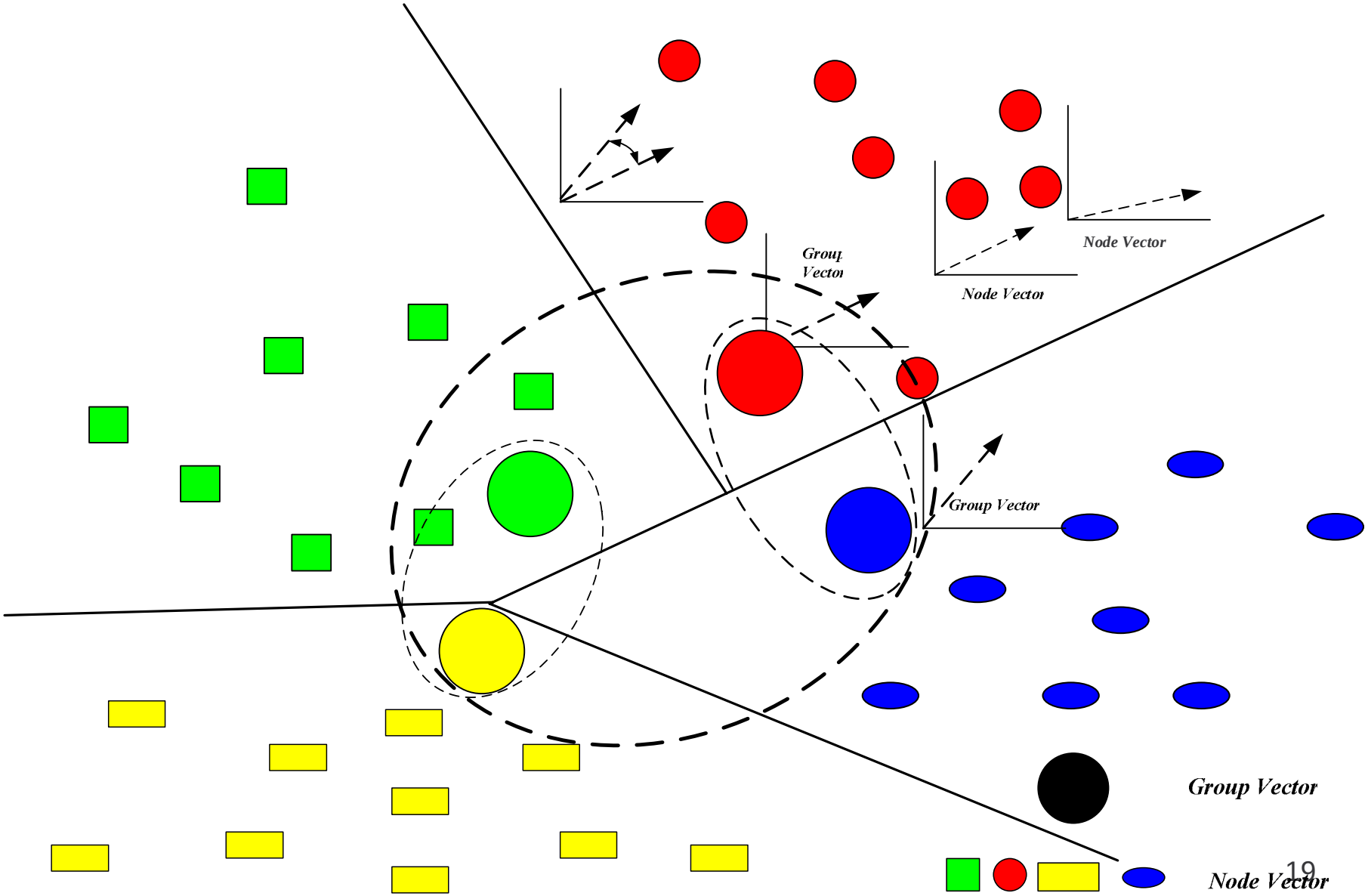
Semantic R-tree

- 构建基于语义特征的扁平化文件命名空间管理机制

Comparisons with Conventional File Systems

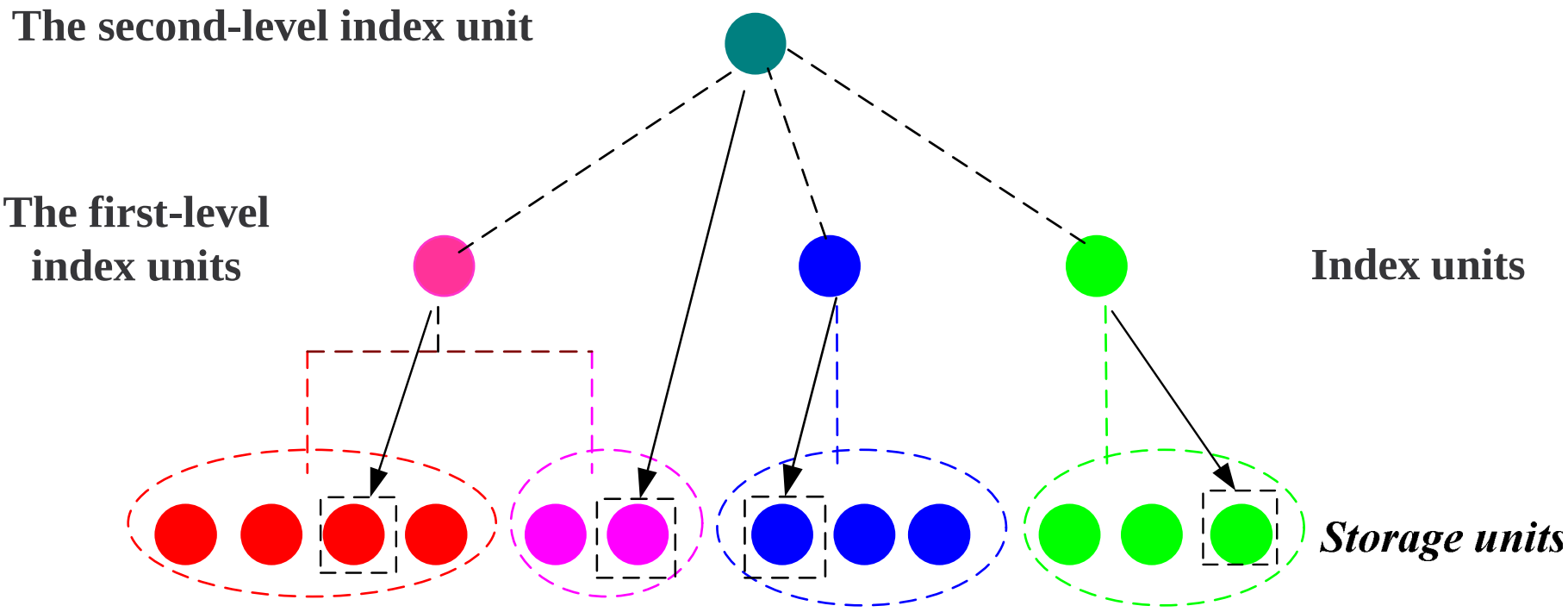


Grouping Procedures

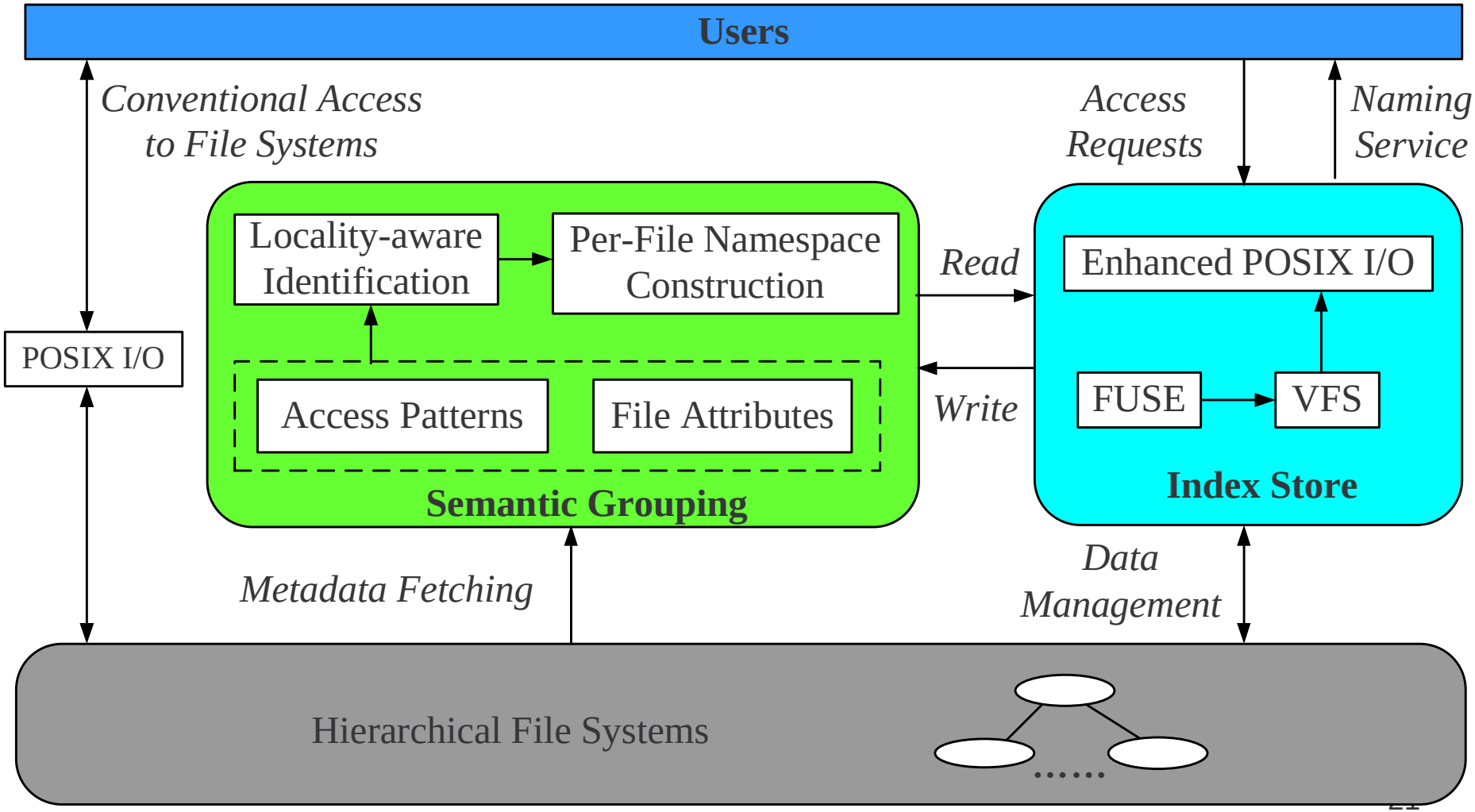


Mapping of Index Units

- **Our mapping is based on a simple bottom-up approach that iteratively applies random selection and labeling operations.**



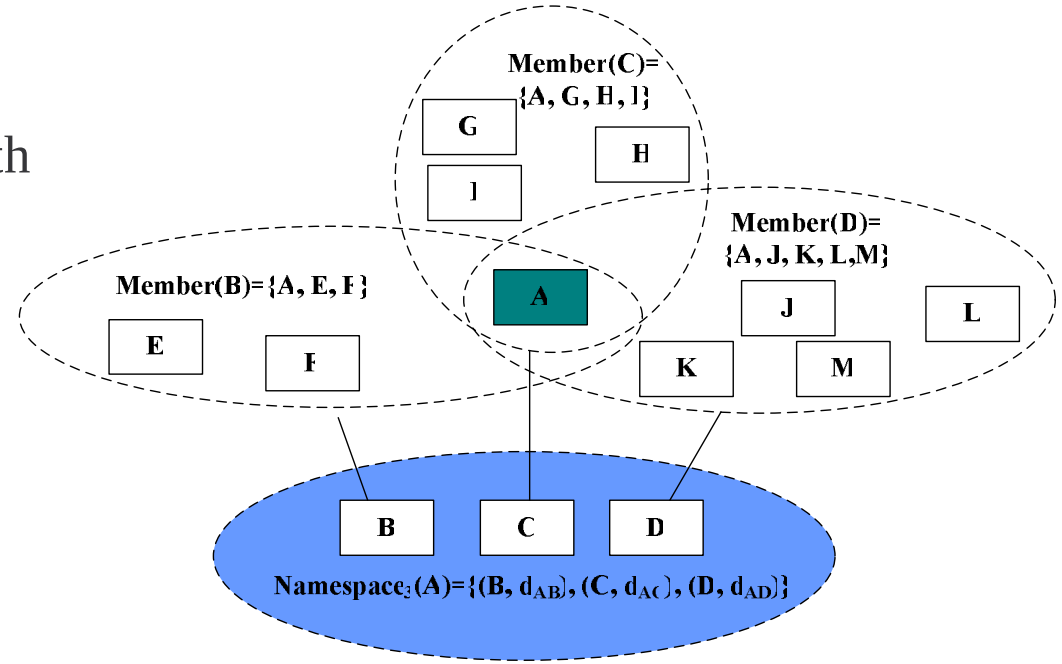
功能组件

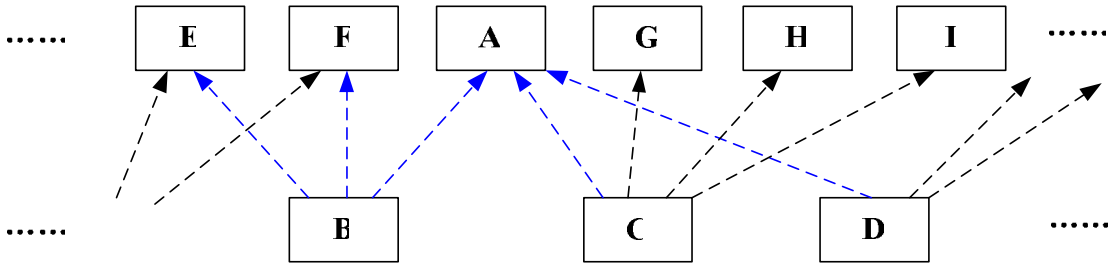

Users
Conventional Access to File Systems
Access Requests
Naming Service
POSIX I/O
Locality-aware Identification
Per-File Namespace Construction
Access Patterns
File Attributes
Semantic Grouping
Read
Write
Enhanced POSIX I/O
FUSE
VFS
Index Store
Metadata Fetching
Data Management
Hierarchical File Systems
.....

Naming and Rename

- **Submodular Maximization**
- Select a subset of namespaces with distinct names

$$S^* \in \underset{S \subseteq V}{\operatorname{argmax}} F(S) \quad s.t. |S| \leq T.$$





Namespace3(A)={(E dAE) (C dAC) (D dAD)}

Member3(B)={(A, dAE), (E, dBE), (F, dBF)}

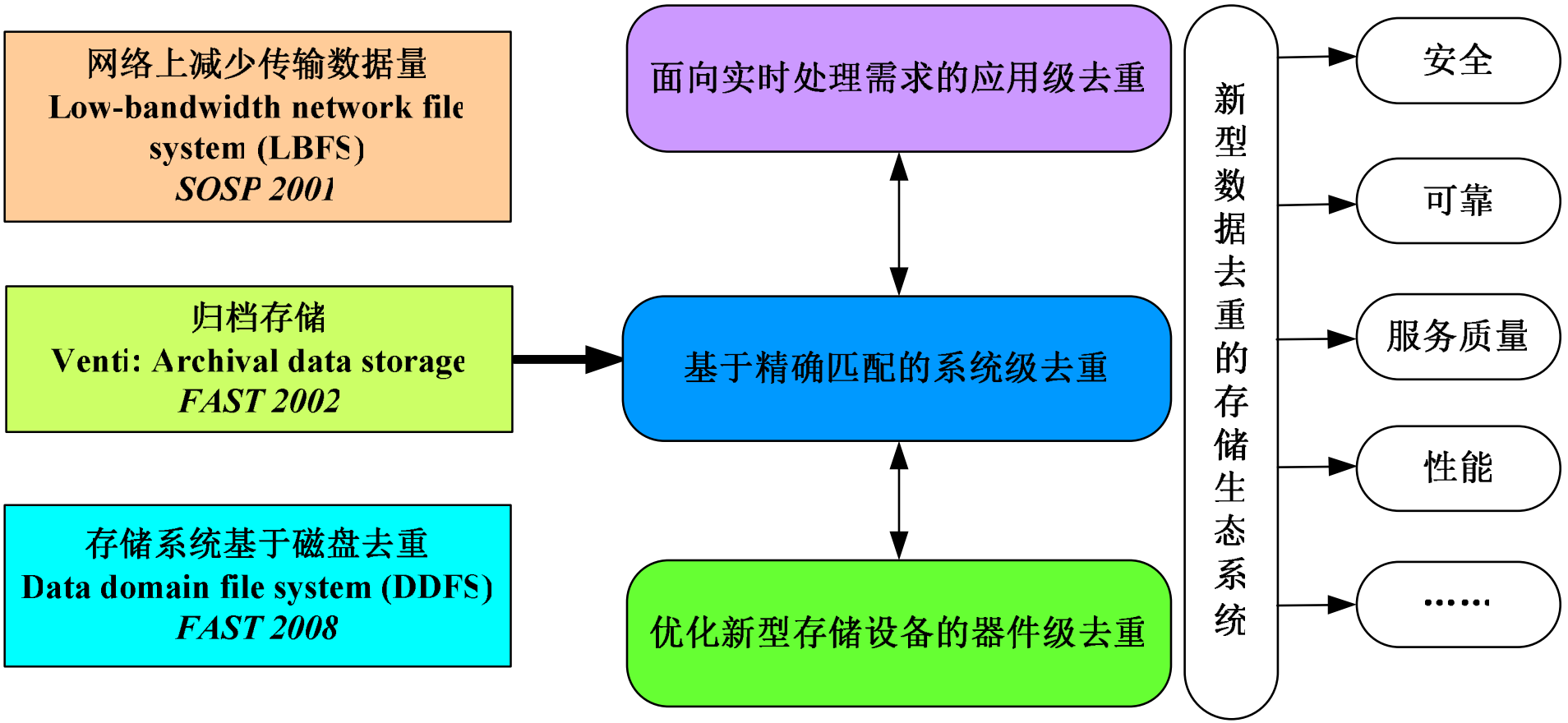
Member4(C)={(A dAC) (G dCG) (H dCH) (I dCI)}

Member5(D)={(A dAD) ····· }

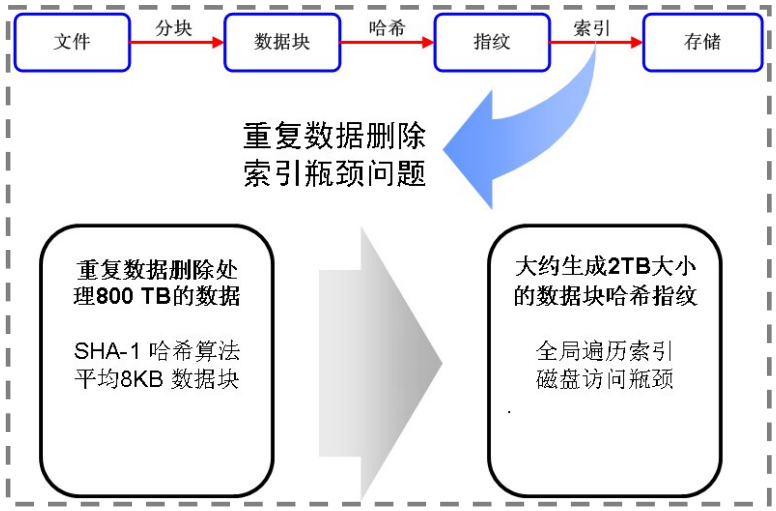
Maximization for Monotone Submodular functions

- Scoring Function is a monotone submodular function
  - Greedy algorithm
  - Constant-scale mathematical quality guarantee

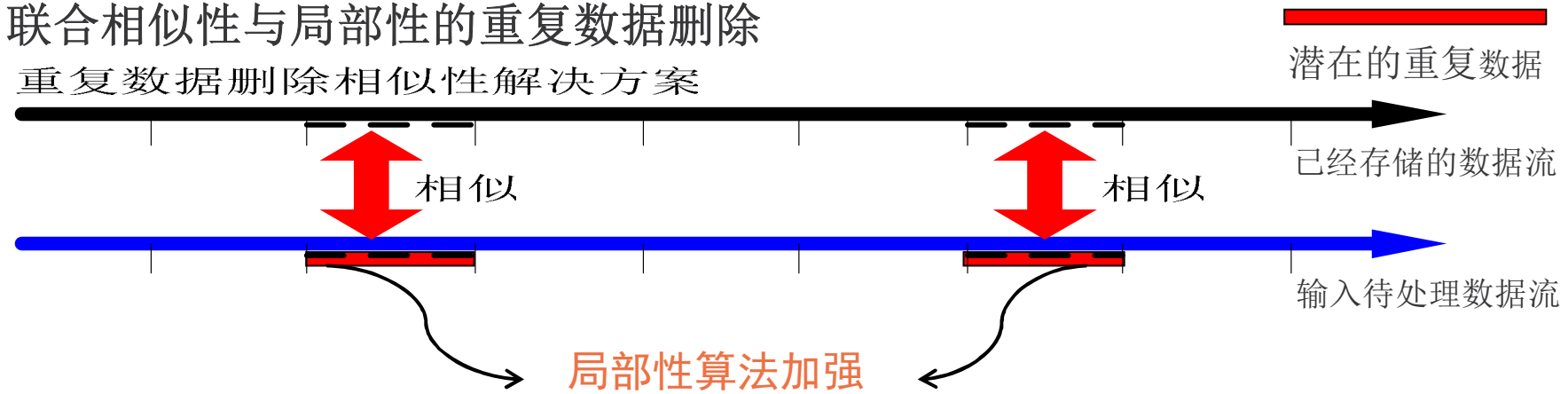
新型系统化的数据去重


网络上减少传输数据量
Low-bandwidth network file system (LBFS)
SOSP 2001
归档存储
Venti: Archival data storage
FAST 2002
存储系统基于磁盘去重
Data domain file system (DDFS)
FAST 2008
面向实时处理需求的应用级去重
基于精确匹配的系统级去重
优化新型存储设备的器件级去重
新型数据去重的存储生态系统
安全
可靠
服务质量
性能
……

联合相似性与局部性的重复数据删除——SiLo



提出并研究基于“相似性与局部性结合”的重复数据删除方法，从理论和关键技术层面研究重复数据删除索引方法，通过充分挖掘备份数据流的相似性与局部性，来改进重复数据删除的整体性能。

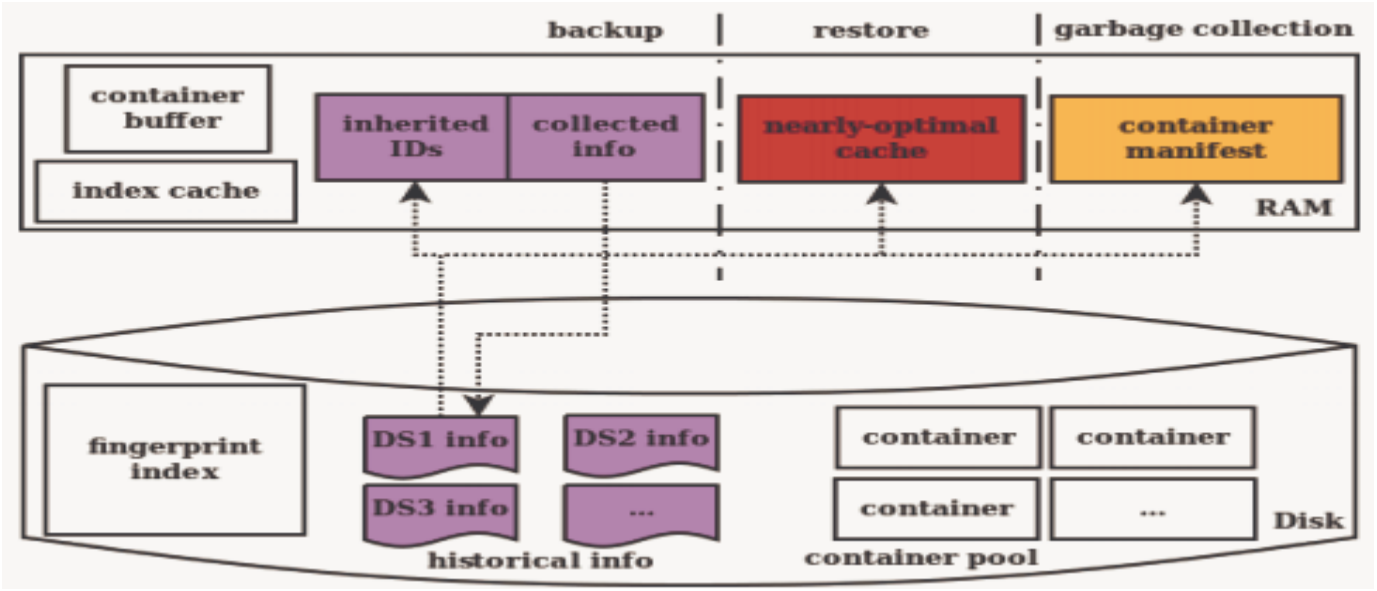


- 通过对相似性的挖掘，避免了全局遍历索引
- 通过对局部性的挖掘，补充相似性查找效果

*“SiLo: A Similarity-Locality based Near-Exact Deduplication Scheme with Low RAM Overhead and High Throughput,” Proceedings of **USENIX ATC**, June 2011.*

数据去重系统的去碎片化算法研究

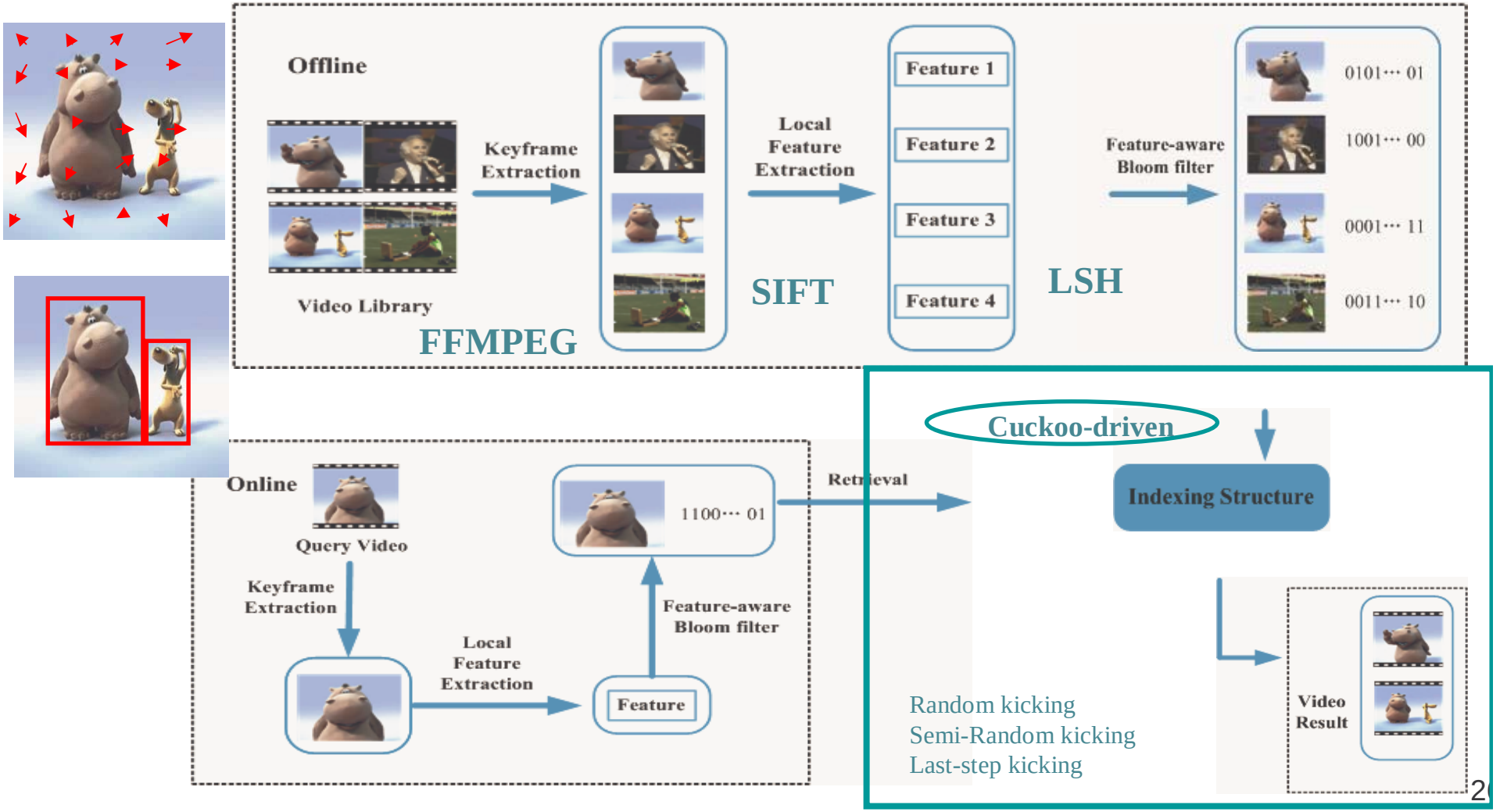
- 提出历史信息感知的重写算法HAR



- **数据去重系统的碎片化严重影响了恢复和垃圾回收效率，研究工作发现稀疏容器是造成碎片化的主要原因**
- **提出基于历史信息感知的重写算法HAR，HAR利用稀疏容器的继承性，准确地找到并重写稀疏容器，非常有效地改善恢复性能和垃圾回收效率**

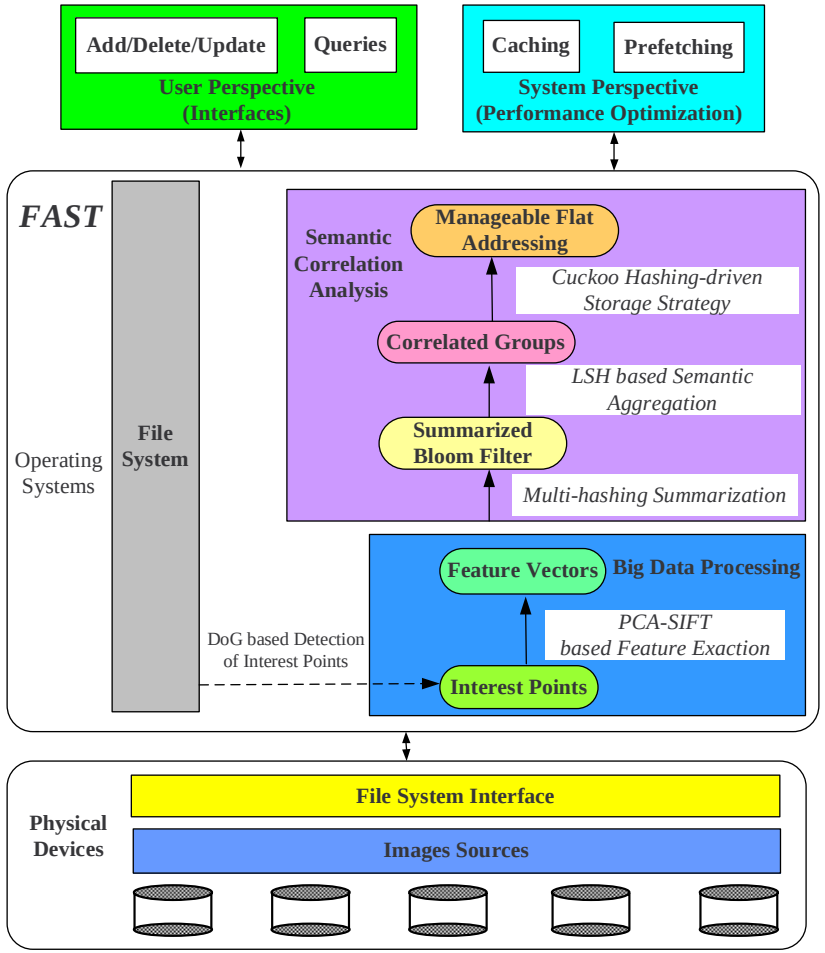
*"Accelerating Restore and Garbage Collection in Deduplication-based Backup Systems via Exploiting Historical Information", Proc. **USENIX ATC**, 2014,*

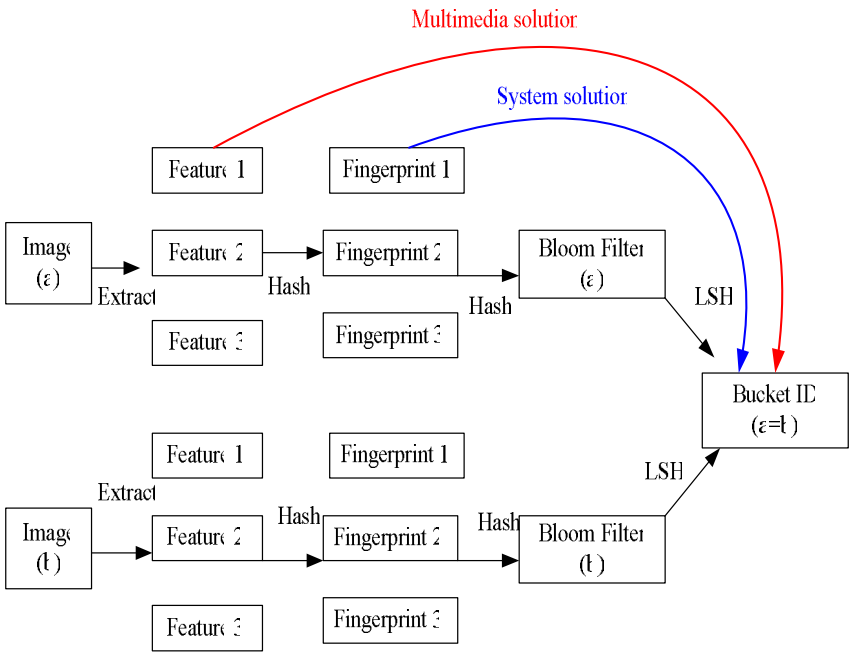
应用级近似去重的方法论：FAST



*"FAST: Near Real-time Searchable Data Analytics for the Cloud", Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (**SC**), November 2014*

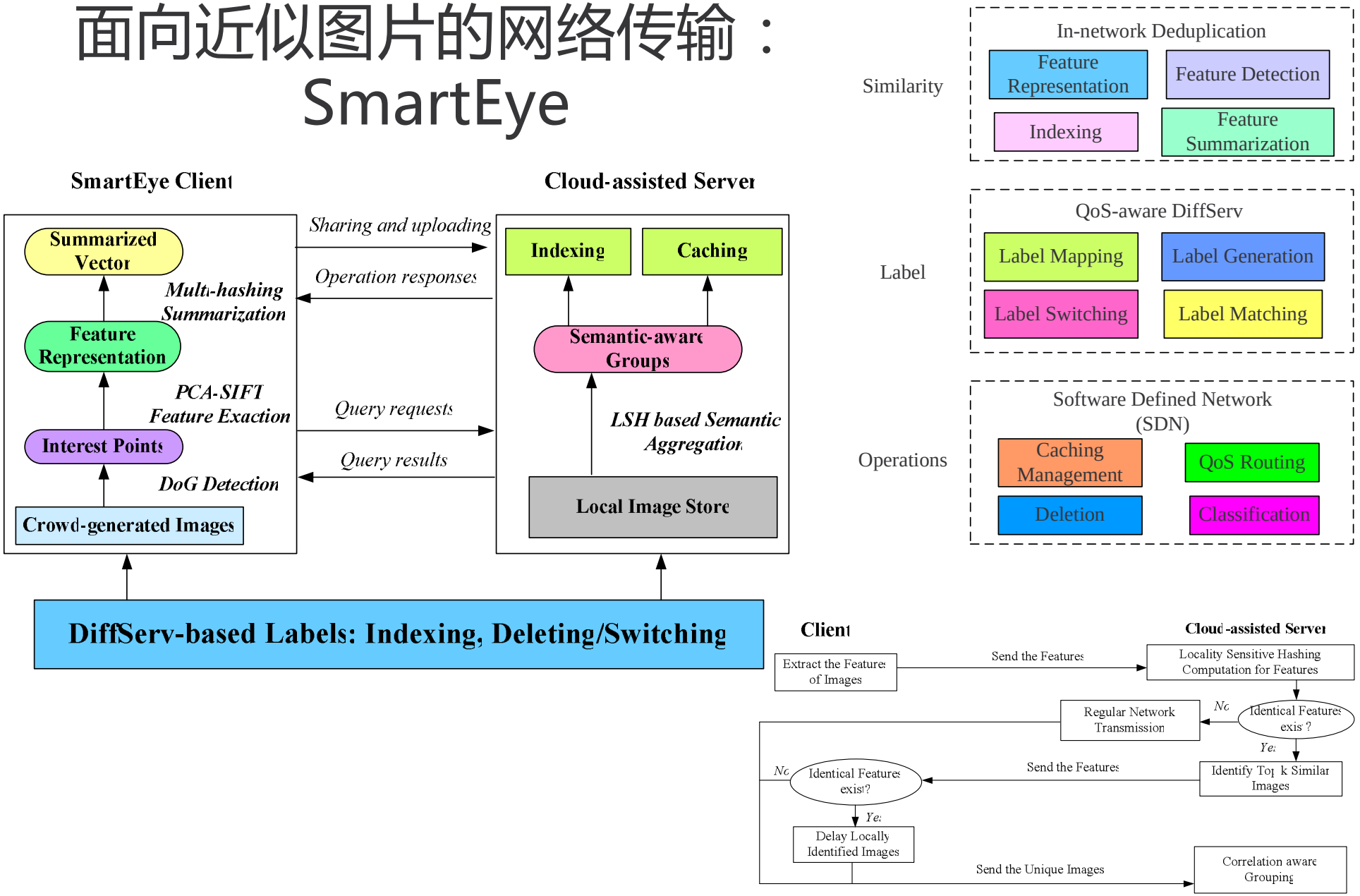
应用级近似去重的方法论：FAST





*"FAST: Near Real-time Searchable Data Analytics for the Cloud", Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (**SC**), November 2014*

面向近似图片的网络传输：SmartEye



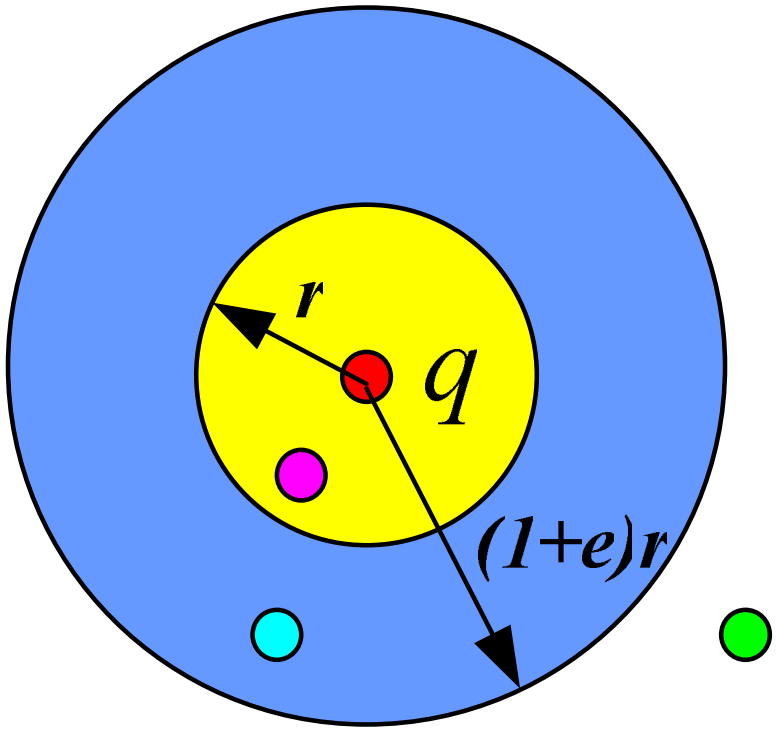
"SmartEye: Real-time and Efficient Cloud Image Sharing for Disaster Environments" Proceedings of INFOCOM, 2015, pages: 1616-1624

Locality Sensitive Hashing (LSH)

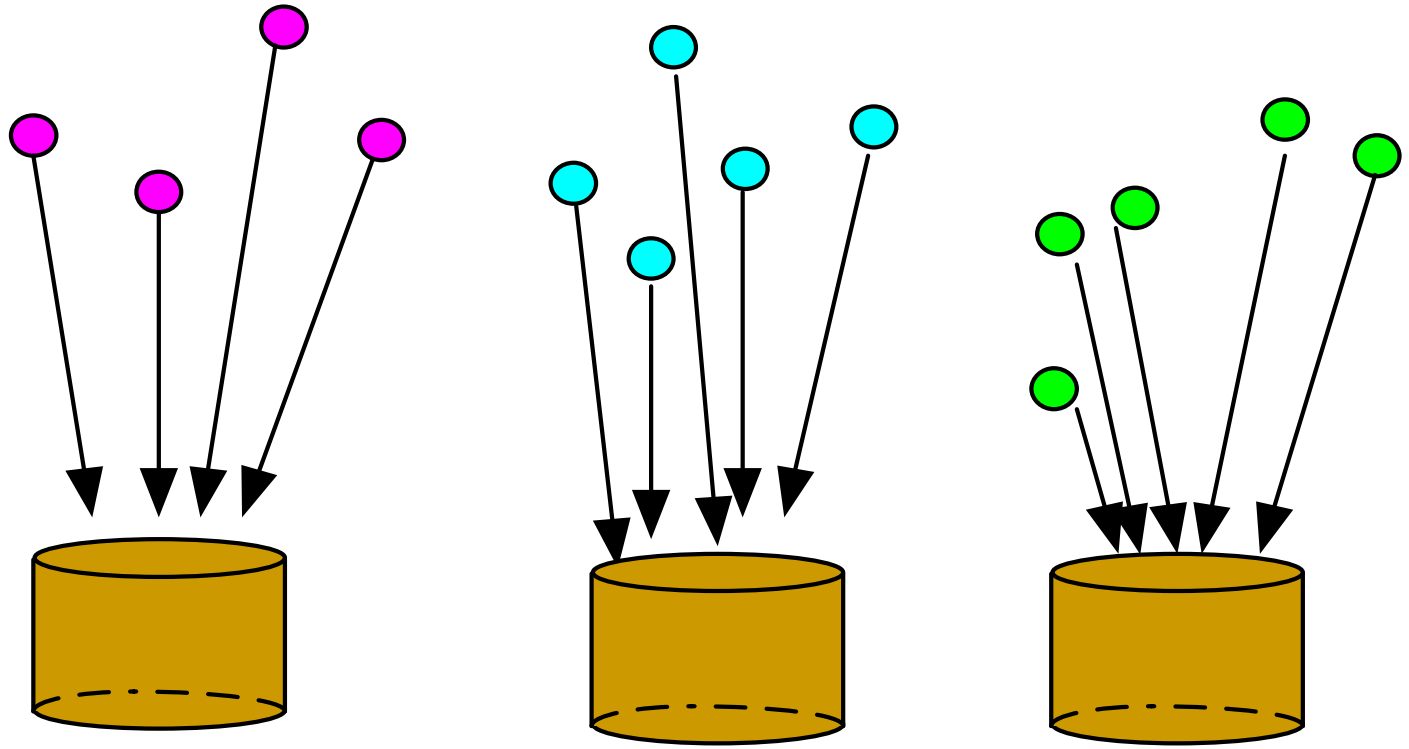
- If $||p,q||_s \leq R$ then $Pr_{\mathbb{H}}[h(p) = h(q)] \geq P_1$,
- If $||p,q||_s > cR$ then $Pr_{\mathbb{H}}[h(p) = h(q)] \leq P_2$.

Near neighbor?

- ***yes***
- ***not sure***
- ***no***



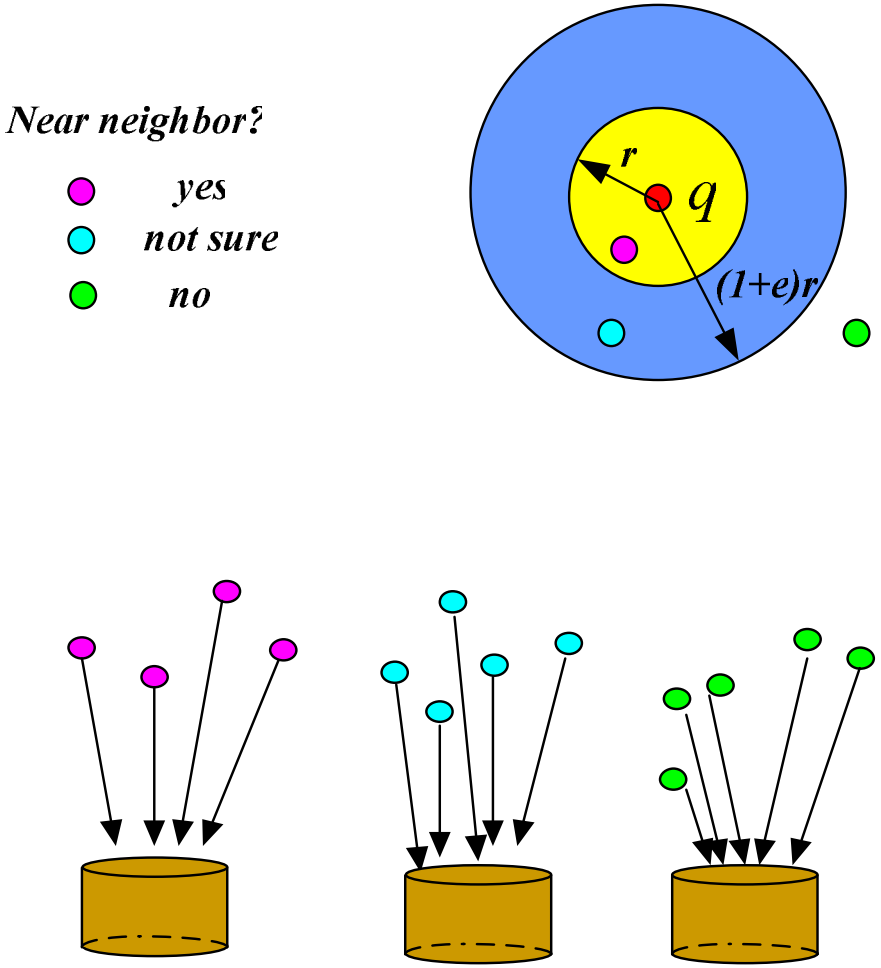
Locality-Sensitive Hashing (LSH)



- **Close items will collide with high probability**
- **Distant items will have very little chance to collide**

NEST: 面向关联感知的近似查询

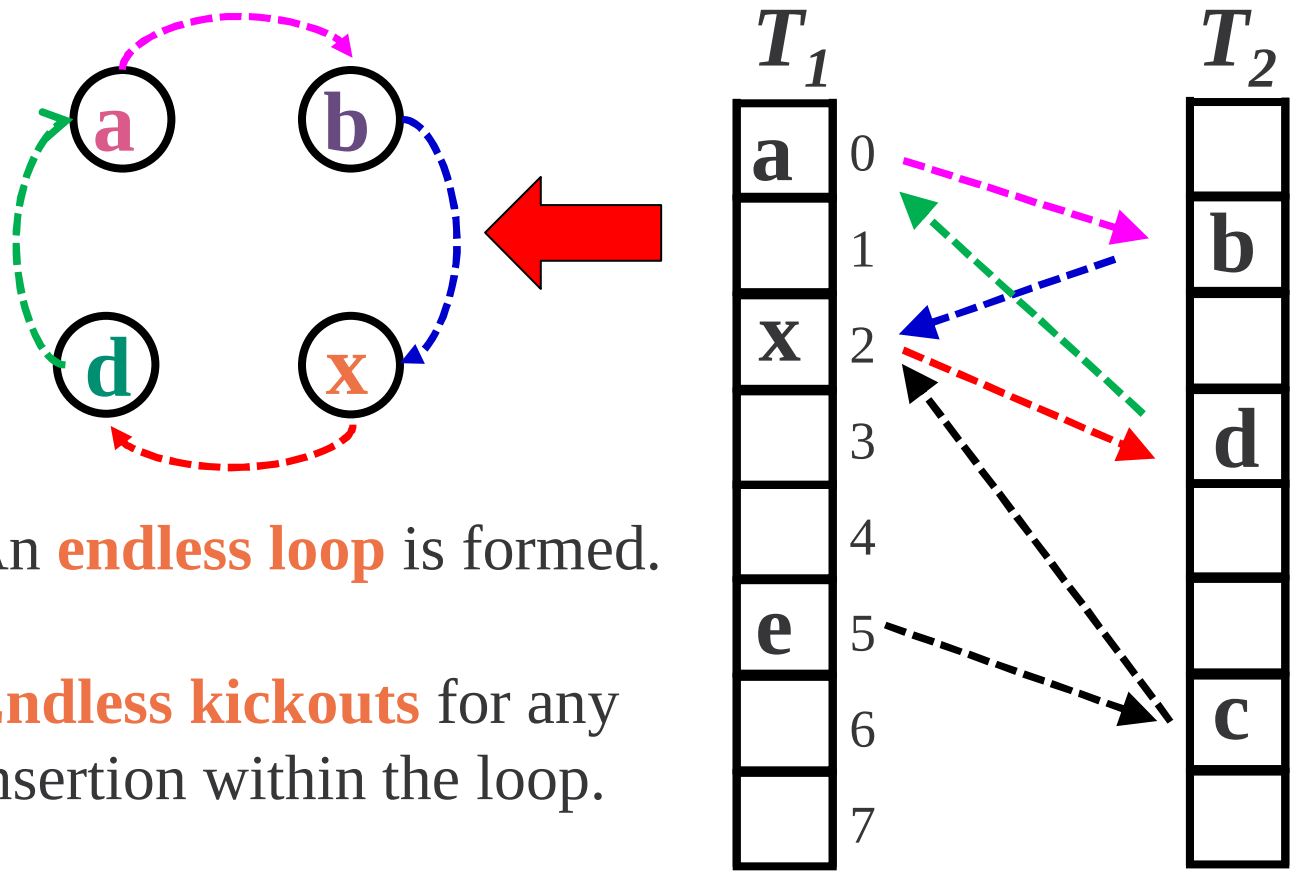
- 近似查询服务于云存储环境中海量、异构、动态和不确定的数据
- 通过哈希计算实现常数量级的快速分类
- 挖掘和获取数据的语义特征和行为模式
- 提高查询服务质量，减少空间负载



"NEST: Locality-aware Approximate Query Service for Cloud Computing", Proceedings of INFOCOM, April 2013, pages: 1327-1335

*"DLSH: A Distribution-aware LSH Scheme for Approximate Nearest Neighbor Query in Cloud Computing", Proceedings of ACM Symposium on Cloud Computing (**SoCC**), 2017*

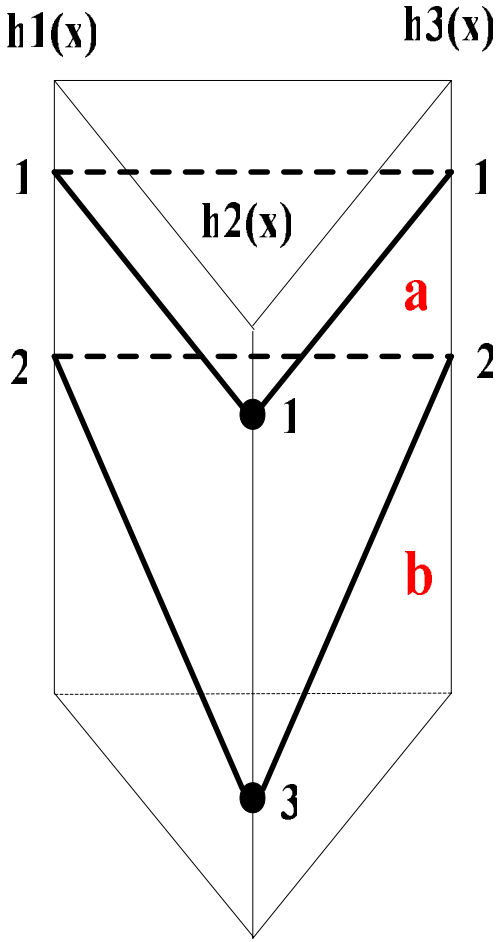
Pseudoforest

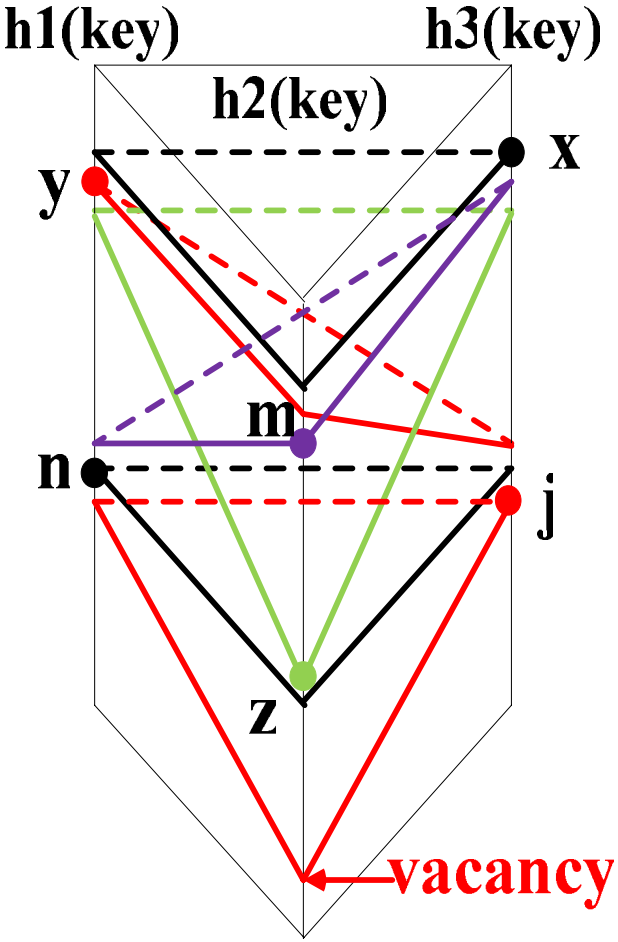


- An **endless loop** is formed.
- **Endless kickouts** for any insertion within the loop.

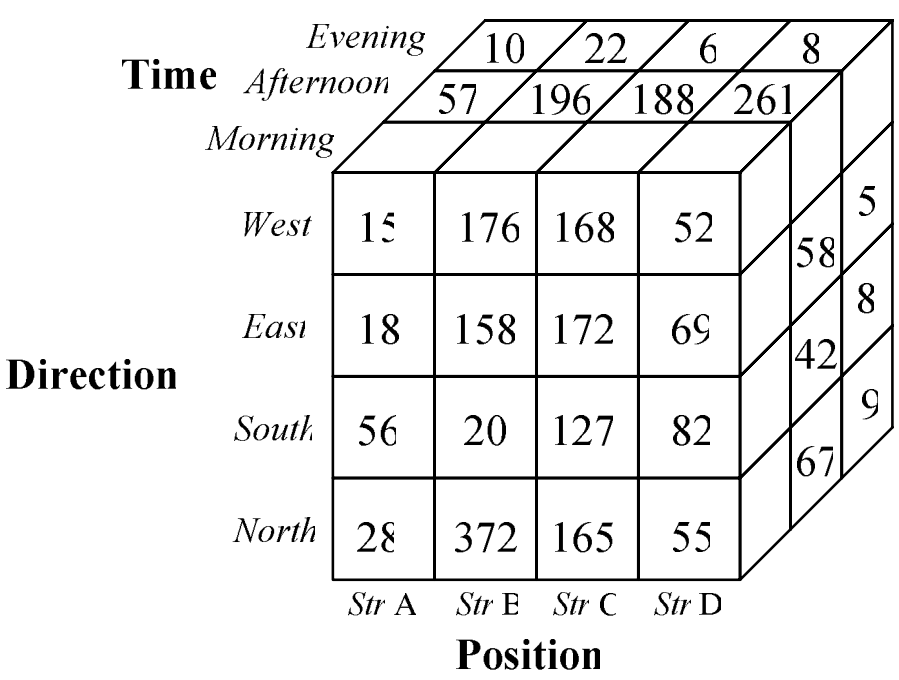
*"SmartCuckoo: A Fast and Cost-Efficient Hashing Index Scheme for Cloud Storage Systems", Proceedings of USENIX Annual Technical Conference (**USENIX ATC**), July 2017, pages: 553-566*

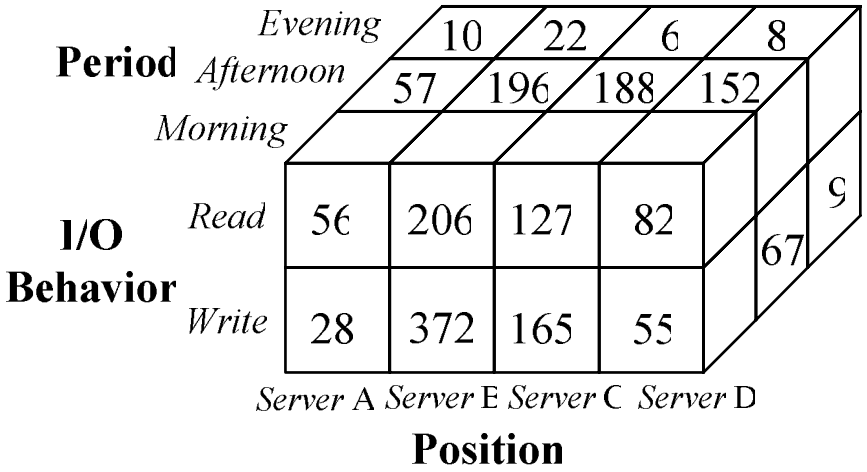
Active prefetching



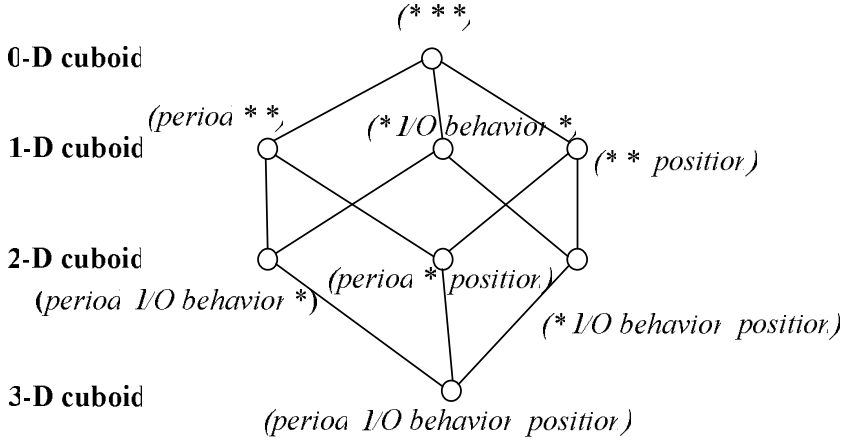


在线的预计算模式：Data Cube





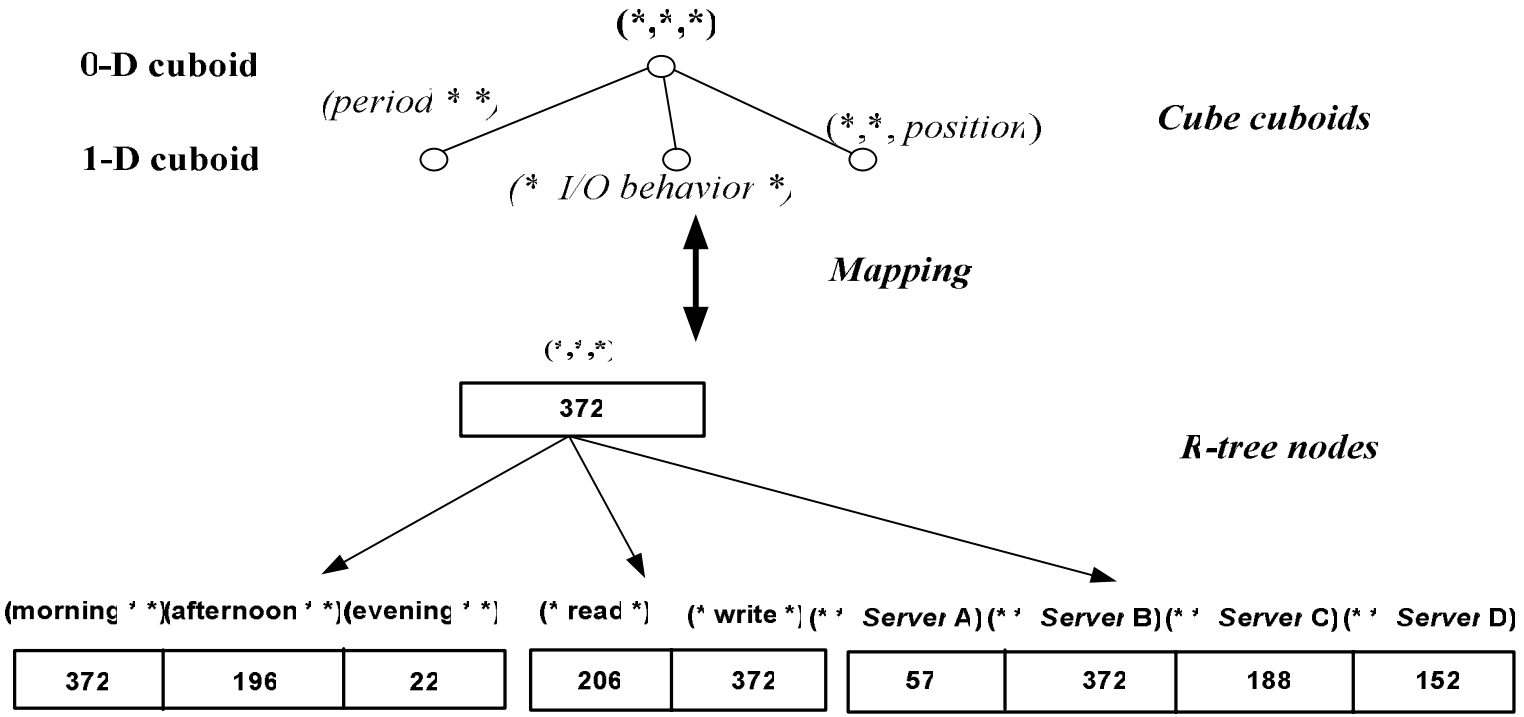
- 数据规模不断扩大
- 获取数据的知识越来越困难
- 提出基于统计信息的数据预计算的方法，提供数据分析服务



"ANTELOPE: A Semantic-aware Data Cube Scheme for Cloud Data Center Networks", IEEE Transactions on Computers (TC), Vol.63, No.9, September 2014, pages: 2146-2159.

语义感知的数据立方体ANTELOPE: 数据映射和存储结构

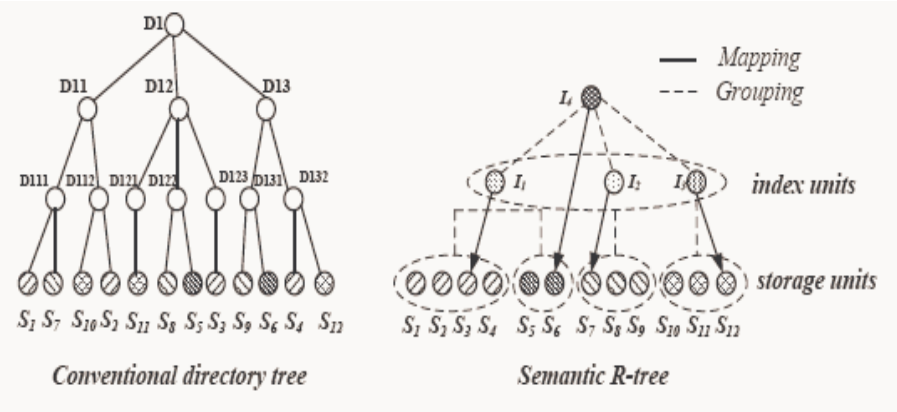
- 对于数据立方体按照数据的多维属性特征进行映射
- 每个特征节点具有多维的范围信息

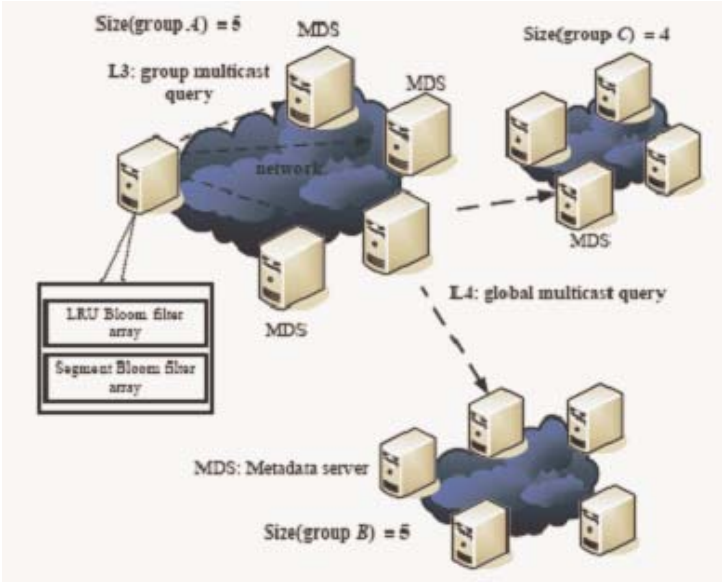


研究工作一：语义感知的存储组织模式

技术难题：存储器件容量受限，数据海量且异构
创新点：感知数据语义，实现关联存储

- 挖掘数据多维属性的语义特征
- 构建关联文件在相同或相近的组内
- 实现语义组织模式和扁平化命名空间
- 与国际前沿方案相比，空间开销平均下降了41.25%，时间延迟平均下降21.6%。

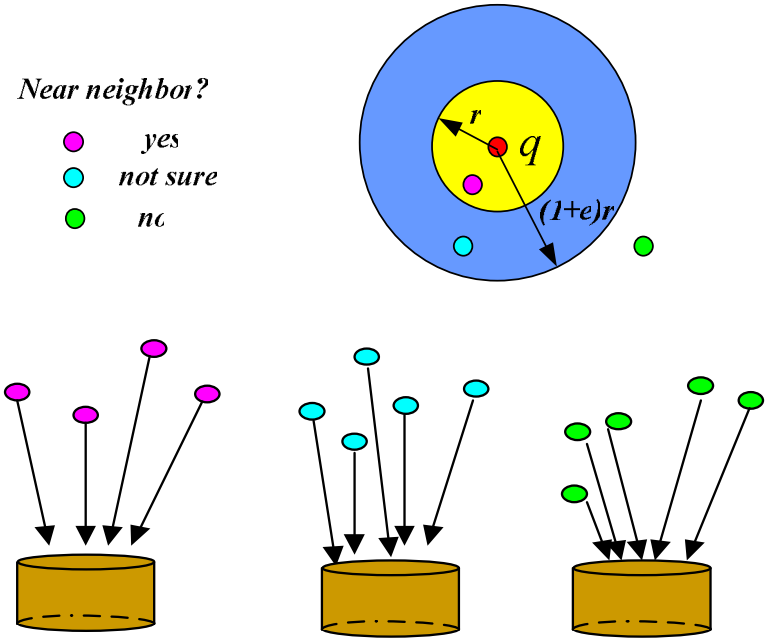




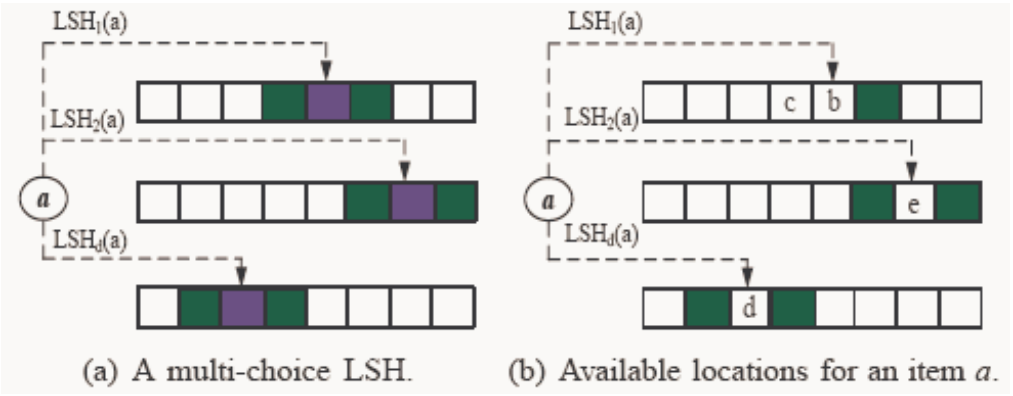
研究工作二：超快哈希计算的数据布局

技术难题：存储服务要求和所需的数据隔离
创新点：面向多维特征，实现供需融合

- 支持近似查询的局部性敏感的哈希结构
- cuckoo驱动和相邻放置实现哈希表的负载平衡
- 实现0(1)复杂度的扁平寻址，优于传统0(n)复杂度的垂直寻址
- 所需存储空间为前沿方法的36%～57%，查询准确性高8%以上



- If $||p,q||_s \leq R$ then $Pr_{\mathbb{H}}[h(p) = h(q)] \geq P_1$,
- If $||p,q||_s > cR$ then $Pr_{\mathbb{H}}[h(p) = h(q)] \leq P_2$.

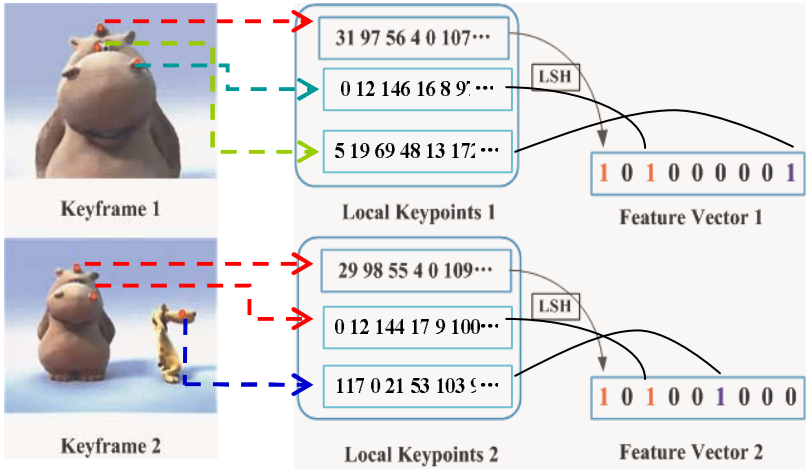


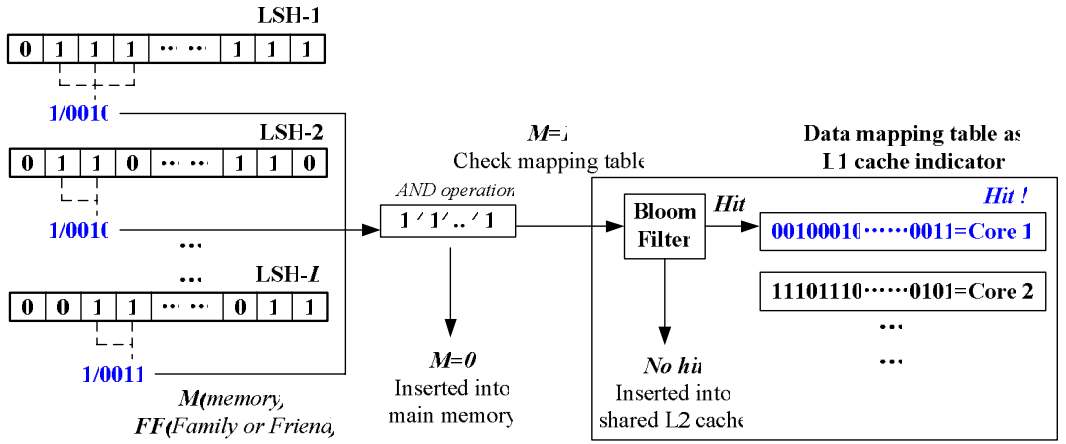
(a) A multi-choice LSH. (b) Available locations for an item *a*.

研究工作三：语义降维的数据去重

技术难题：数据冗余、海量和低质
创新点：轻量级的超级特征值，实现存算联动

- 多维特征的轻量级压缩
- 异构成员关系的快速编码，比特级的快速比较
- 2百万图片的去重查询从原有的12分钟降低到1秒以内

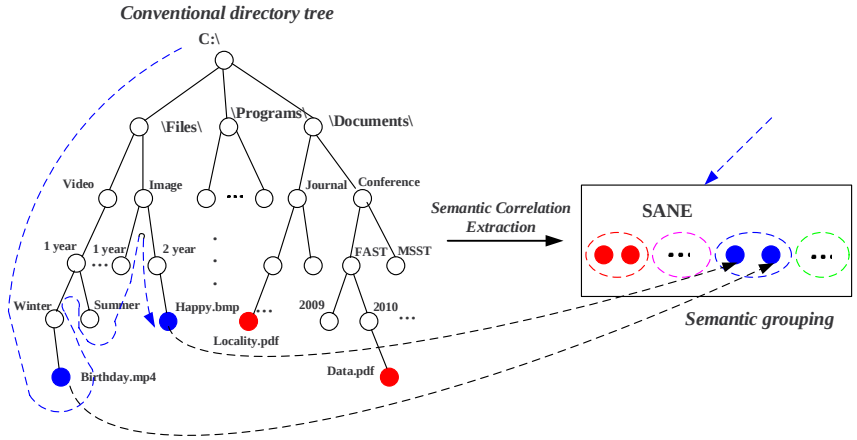


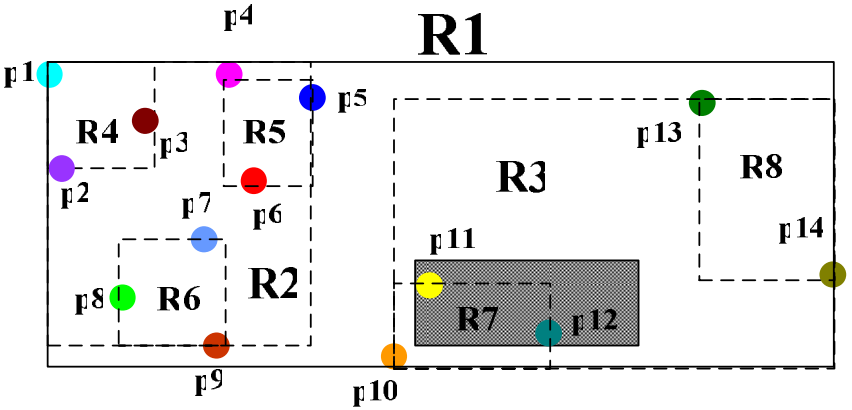


总结

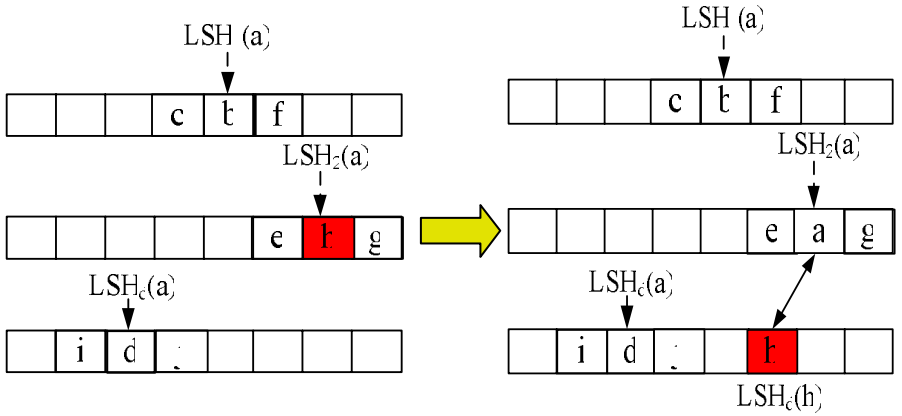
与传统的存储模式和数据布局方法比较

➢结构：由层次化向扁平化转变

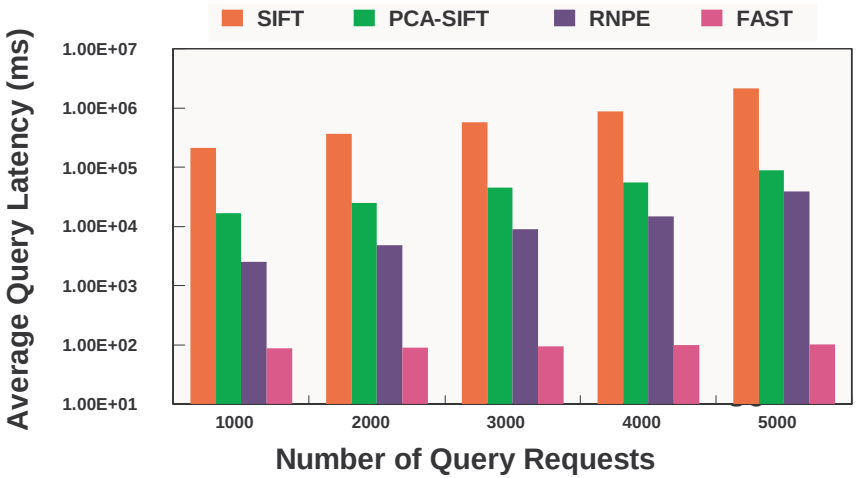




➢算法：由难解向易解转变



100倍的性能提升



Open Source Codes (in GitHub)

- ***SmartCuckoo****: in GitHub. SmartCuckoo is a new cuckoo hashing scheme to support metadata query service.*
  - **https://github.com/syy804123097/SmartCuckoo**
- ***SmartSA*** *(E-STORE)****:*** *in GitHub to support near-deduplication for image sharing based on the energy availability in Smartphone.*
  - **https://github.com/Pfzuo/SmartSA**
- ***Real-time-Share****: in GitHub, to support real-time image sharing in the cloud, which is an important component of SmartEye (INFOCOM 2015).*
  - **https://github.com/syy804123097/Real-time-Share**
- ***MinCounter****: in GitHub. MinCounter is the proposed data structure in the MSST 2015 Paper.*
  - **https://github.com/syy804123097/MinCounter**
- ***NEST****: in GitHub (Download INFOCOM 2013 Paper, Source Codes, Manual and TraceData).*
  - **https://github.com/syy804123097/NEST**
- ***LSBF*** *(Locality-Sensitive Bloom Filter): in GitHub (Download TC 2012 Paper, Source Codes and Manual).*
  - **https://github.com/syy804123097/LSBF**

Thanks and Questions