華中科技大學
计算机科学与技术学院
School of Computer Science &Technology，HUST

LoopDelta: Embedding Locality-aware Opportunistic Delta Compression in Inline Deduplication for Highly Efficient Data Reduction

讲解人：莫柠源
时间：2023.11.28

目　录

背景

- 备份系统中的冗余数据

- 重复数据删除（数据去重）

删除重复块，块级粗粒度，开销小，去重率稍低

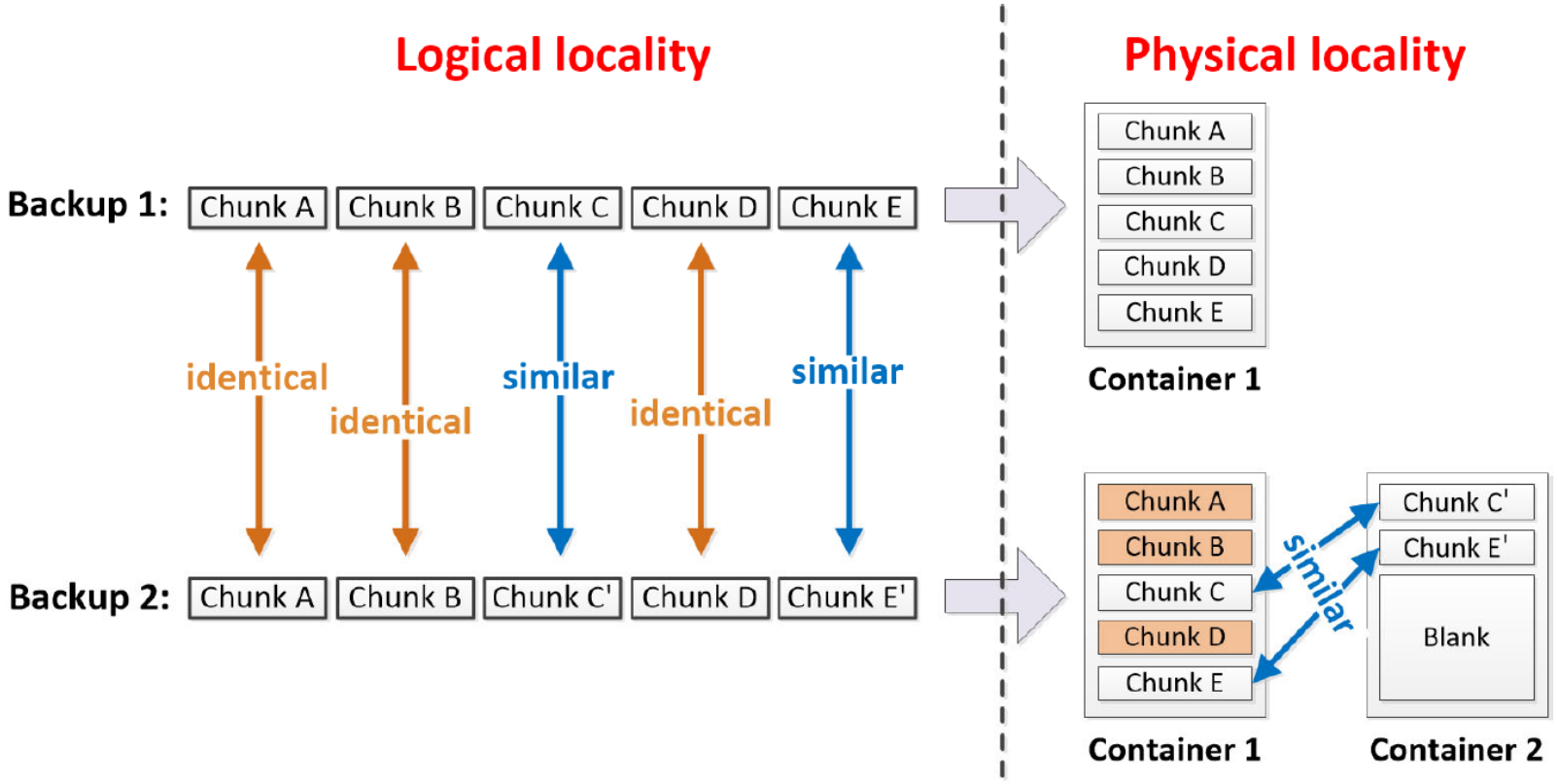
- 差量压缩

移除类似块之间的冗余，细粒度，开销大，去重率高

向重复数据消除系统添加差量压缩的挑战

- **低压缩比**
- **备份吞吐量低**
- **恢复性能低**
- **应用重写技术时丢失潜在相似块**

挑战1：低压缩比



Redundancy Locality：冗余数据在连续备份中的重复模式

- 逻辑位置：重复数据删除前的重复模式
- 物理位置：重复数据删除后的重复模式

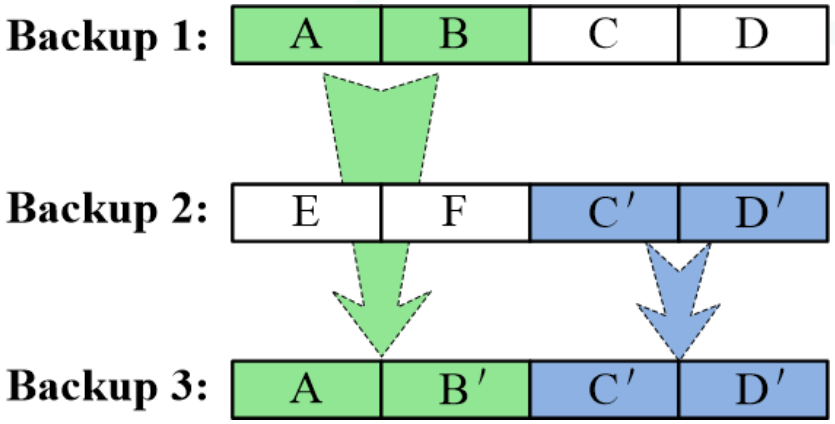
草图索引技术：

- 基于逻辑位置的索引：上次备份的数据块草图
- 基于物理位置的索引：与重复块一起存储的数据块的草图
- 完全索引：备份存储中所有数据块的草图

基于逻辑局部性的草图索引

缺点：备份版本中缺少潜在的类似区块。

优点：检测到的相似组块具有较高的相似性。

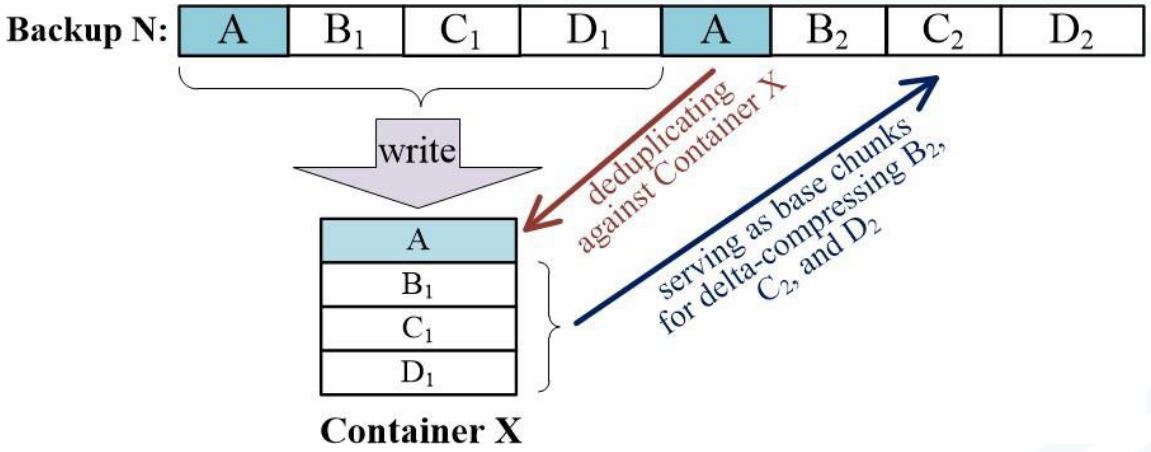


用于差量压缩区块的最佳基本区块通常是其上次备份中的上一个副本

基于物理位置的草图索引

缺点：检测**自参考相似块**作为基本块。

优点：检测大多数潜在相似块。



前序备份中的相似块 > 自参考相似块

完整草图索引

缺点：检测自引用的类似块作为基本块。

优点：检测所有潜在相似块。

压缩评价的上限

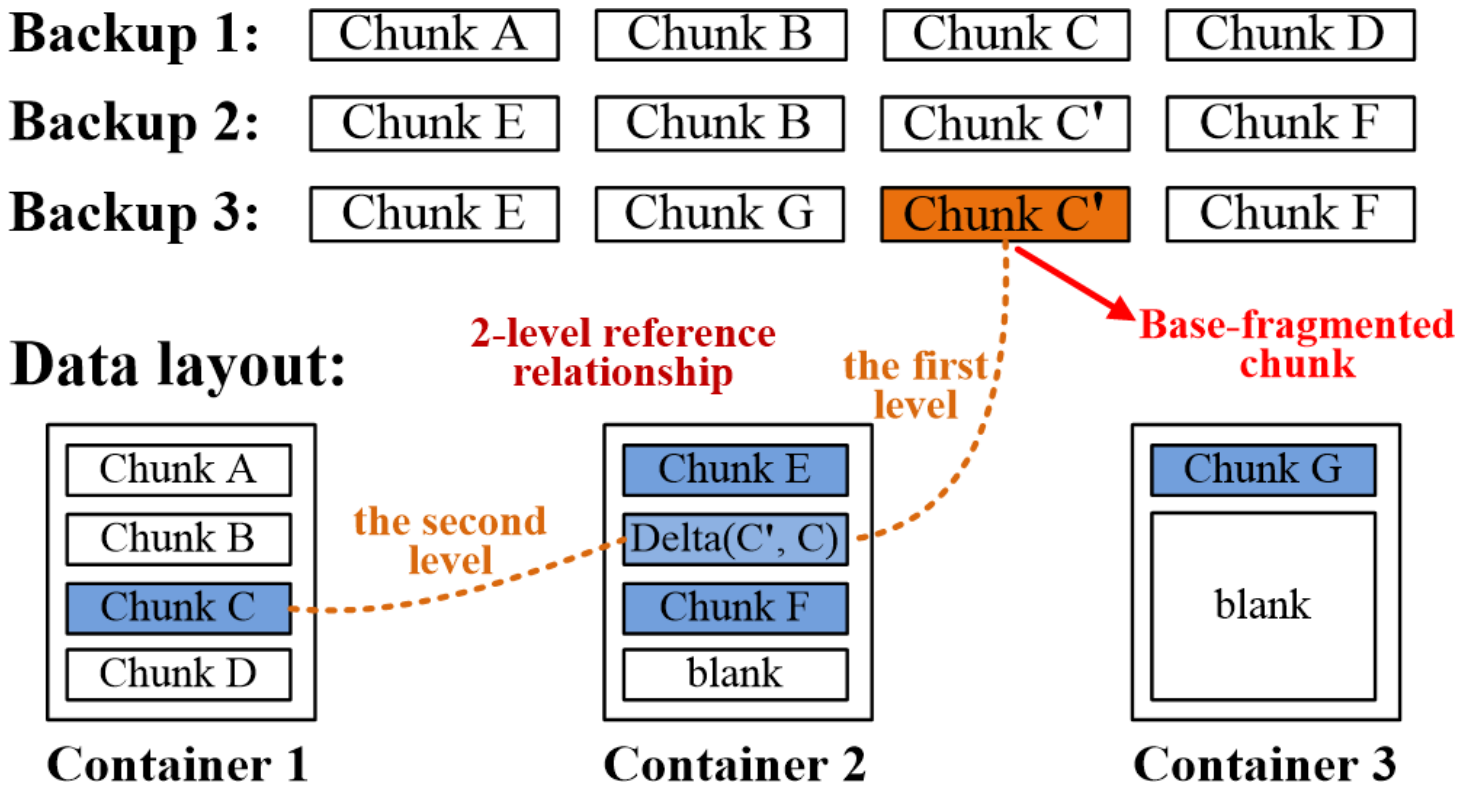
挑战2：备份吞吐量低

用于读取写入路径上的基块的额外 I/O
显著降低备份吞吐量。

观察结果：
- 常规操作：在重复数据删除期间需要访问容器
- 在重复数据删除期间，将访问大多数包含相似数据块的容器

解决方案：位置感知预取

挑战3：恢复性能低



用于在读取路径上读取基块的额外I/O显著降低恢复性能。

- 位置感知预取可减少恢复过程中的额外 I/O。
- **基碎片块**：引用其基块在恢复期间需要额外 I/O 的增量的数据块。

挑战：获取系统中差量的基块的容器ID

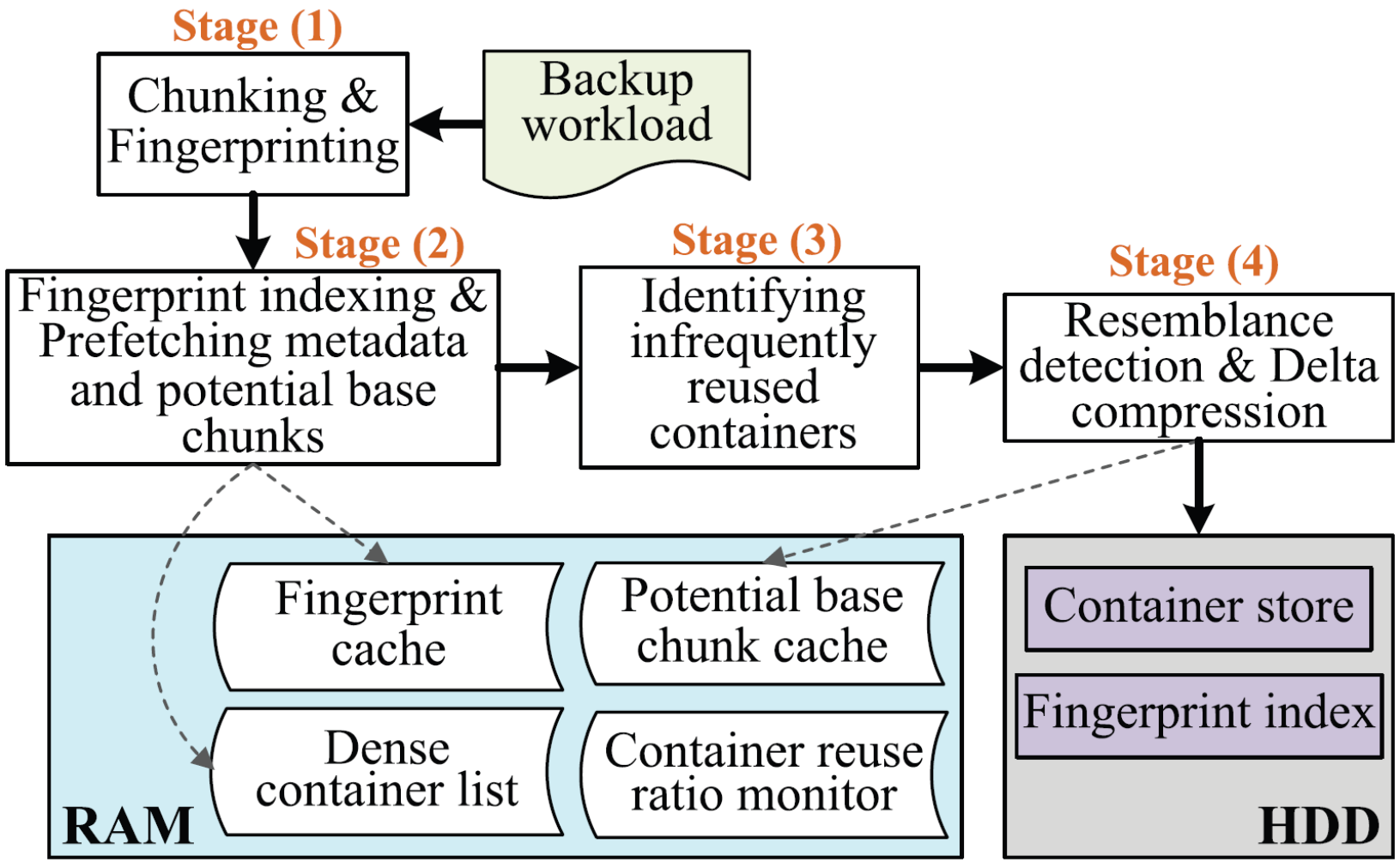
挑战4：缺失基块

- 重写技术识别不频繁重用的容器，减少对其的引用。
- 还原期间需要基块。
- 从不频繁重用的容器中检测到的相似块不能用作基块参与压缩。

观察结果：
差量编码可视为一个两步过程。
- 步骤1：对目标块相对于相似块进行编码并生成增量
- 步骤2：去除目标块并存储差量,实现数据缩减

目标块是指正在备份的区块，而相似块是指备份存储中的区块。

解决方案：反向编码



Loopdelta方案

- 基于双局部性的相似性追踪：通过利用逻辑和物理局部性，检测潜在的相似数据块。
- 局部性感知预取：在备份过程中预取基块，以避免额外的 I/O 操作。
- 缓存感知过滤器：识别以前写入的增量，以消除读取路径中的额外 I/O 操作。
- 反向增量压缩：对那些因重写技术受限的数据块进行增量压缩，从而提高还原性能。

应对1：基于双局部性的草图索引

两者的互补性：

草图	优势	劣势
逻辑位置	高相似性	缺少类似区块
物理位置	检测几乎所有相似的块	低相似性

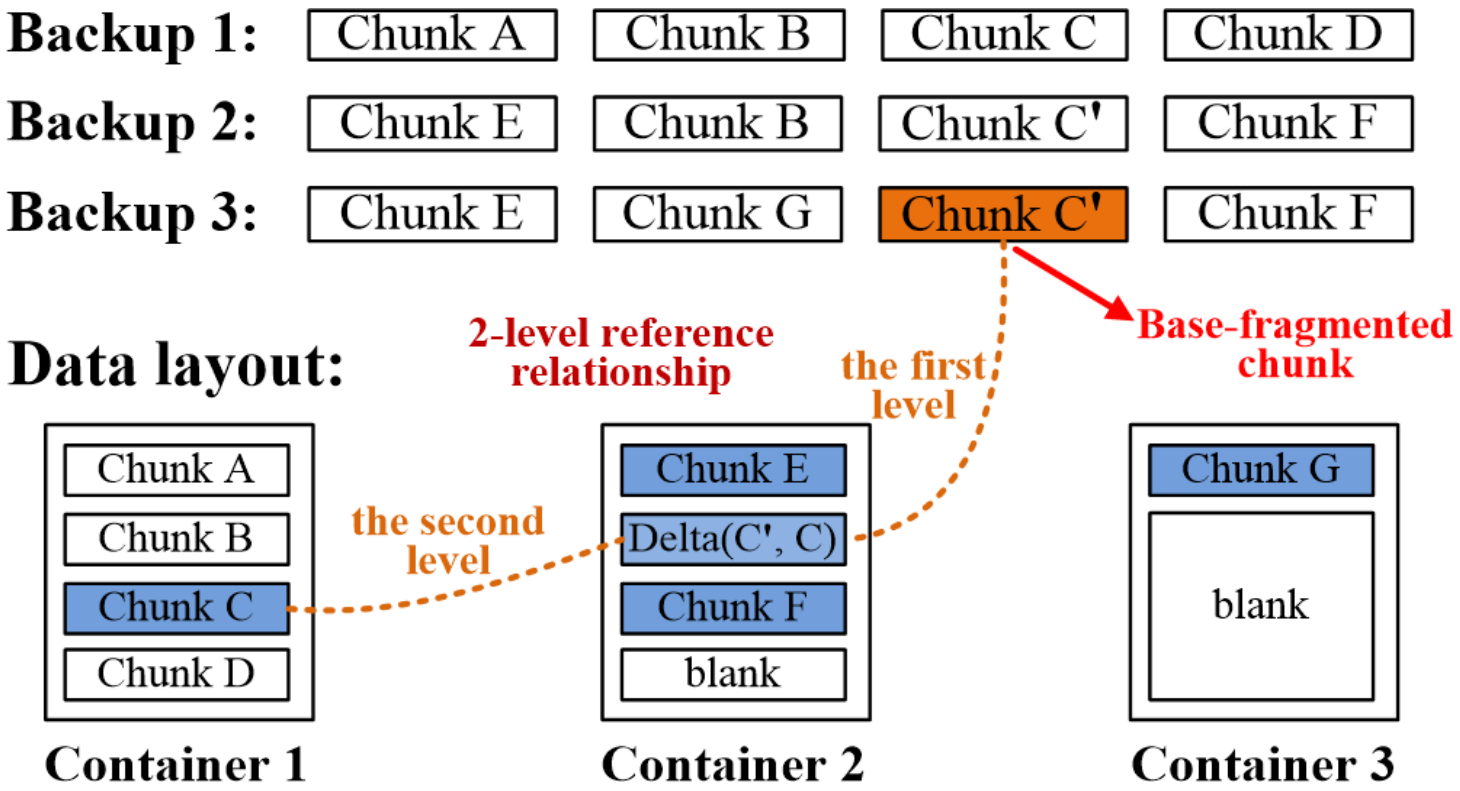
基于双局部性的草图索引：通过利用逻辑和物理局部来检测相似的块

应对2：位置感知预取

- 为指纹高速缓存和Bloom过滤器相结合的盘上指纹索引。
- 重删期间访问的容器，若为上次备份生成的密集容器则预取整个容器，否则仅元数据
- 在GC之后更新密集容器列表来跟踪潜在的基块
- LRU算法逐出

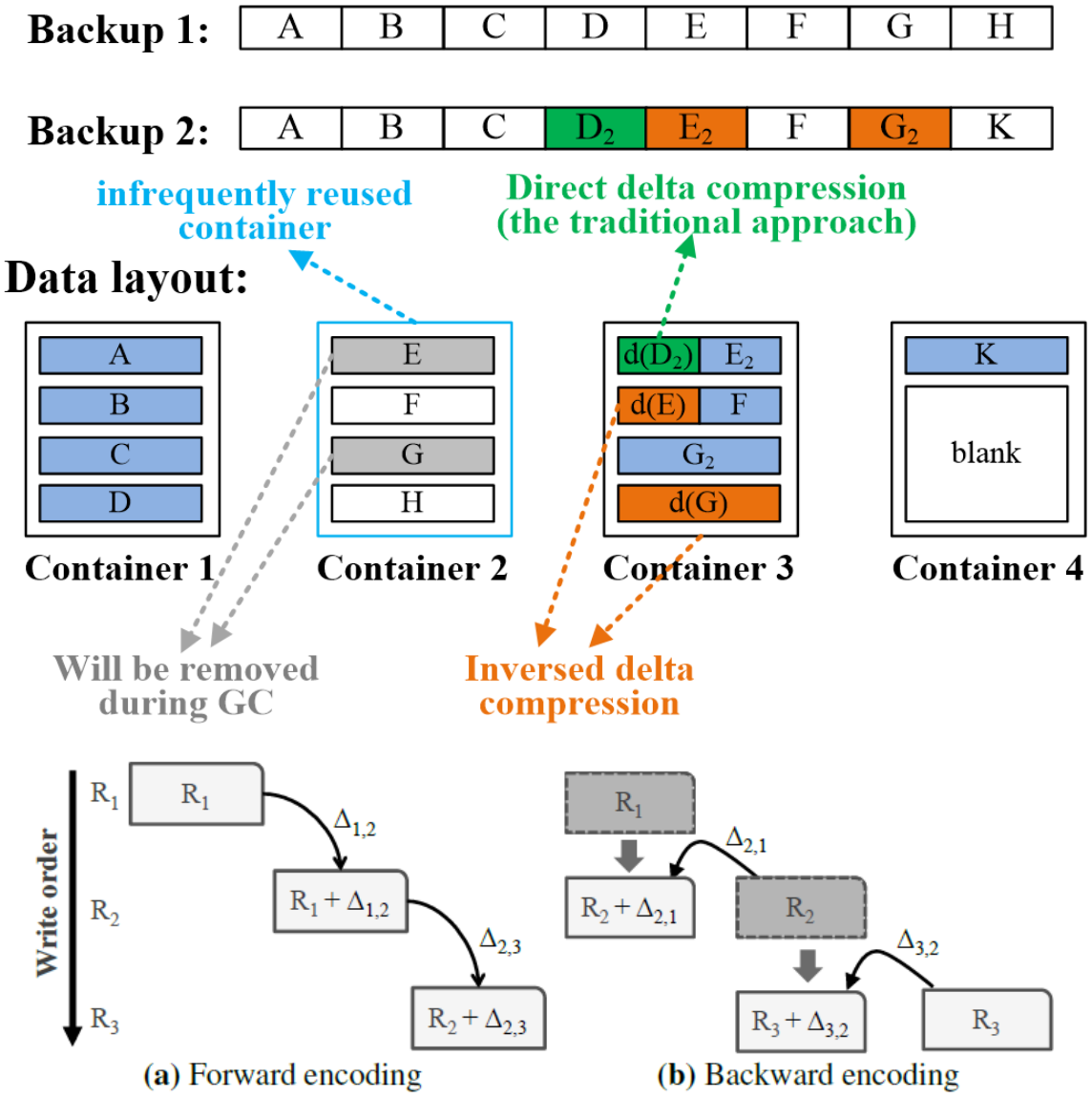
通过在重删期间预取元数据的常规操作上搭载来预取潜在的基块。

应对3：缓存感知筛选器



- 连同增量一起存储基块的指纹。
- 在重复数据删除期间，借助最近预取的元数据识别基块
- 重写基碎片块，以防止在恢复过程中对基块执行额外的 I/O

应对4：反向编码



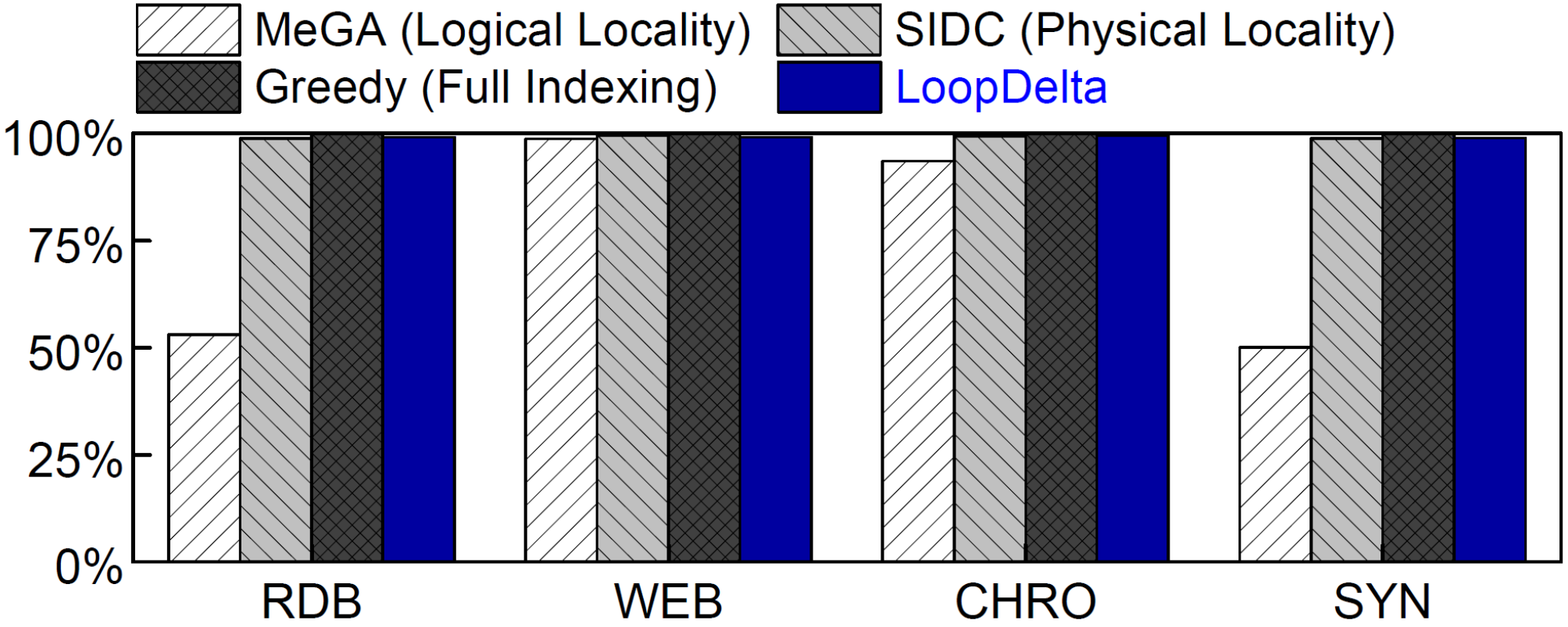
将增量压缩的目标更改为备份中的块储存。

- 步骤1：对检测到的与正在备份的块（例如，C）相似的块（例如，S）进行编码并生成差量,并将该差量与C一起存储。
- 步骤2：在垃圾收集（GC）期间移除S,以实现数据缩减。

评估： 数据集

名称	尺寸	工作负载描述	关键属性
RDB	1080 GB	redis 键值存储数据库的 200 个备份	多版本继承
WEB	330 GB	news.sina.com120天快照	自参考相似块
CHM	284 GB	Chromium项目的100个源代码版本，从v84.0.4110到v86.0.4215	
SYN	335 GB	通过模拟文件创建/删除/修改操作进行180 次合成备份	多版本继承

评估 1： 检测到的相似块百分比

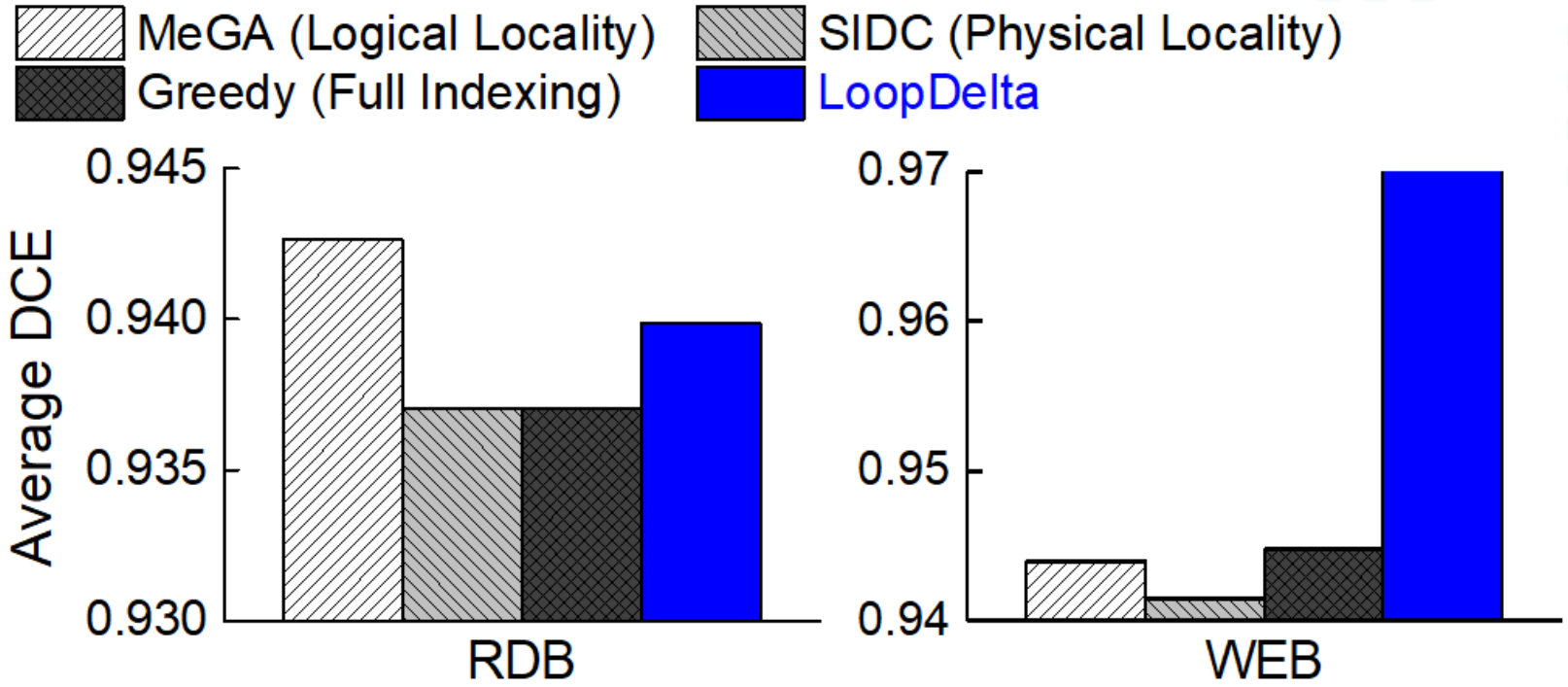


LoopDelta可以检测几乎所有潜在的相似块

评估 2：检测到的组块的相似性

DCE值越大表示相似性越高

$$DCE = \frac{size\ of\ delta-compressed\ bytes}{chunk\ size\ before\ delta\ compression}$$



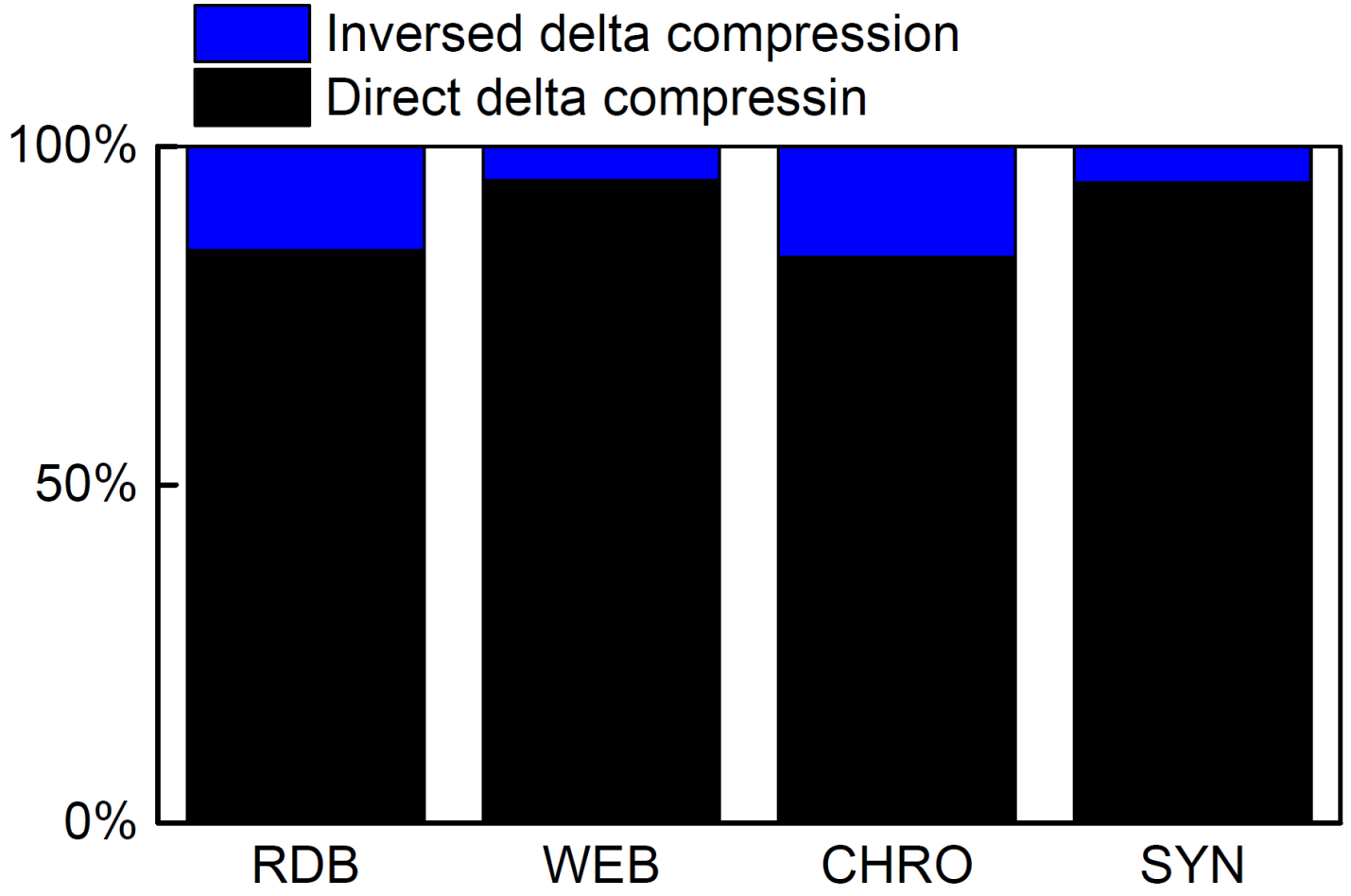
在包含自引用的相似块的数据集（WEB）上，检测到具有比其他方法更高的相似性的相似块。

评估 3： 局部性感知预取性能

数据集	改善（%）
RDB	50.6%
WEB	11.7%
CHRO	33.3%
SYN	47.8%

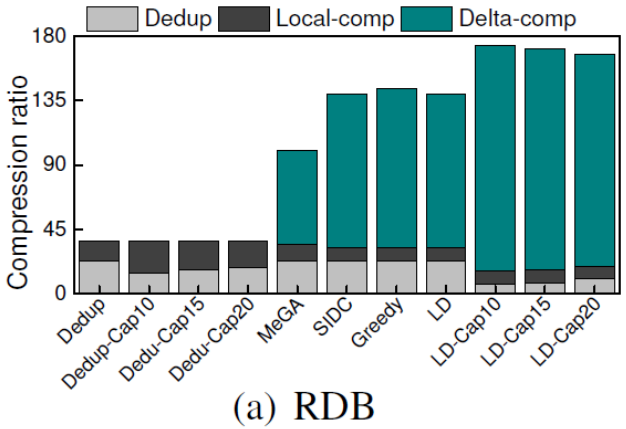
在应用重写时，通过高速缓存感知过滤器实现的恢复性能的改进

评估 4： 反向编码的有效性

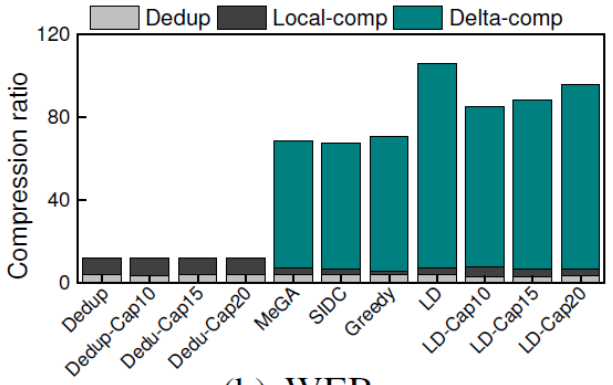


- 重写方案已封顶
- 压缩增益：15.3%、5%、16.4%和5.3%

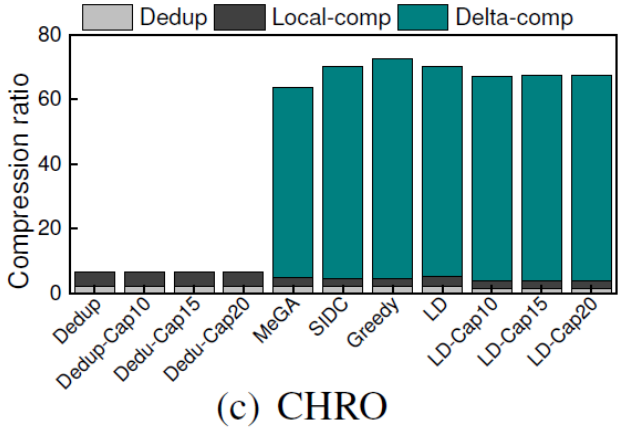
评估 5： 压缩比



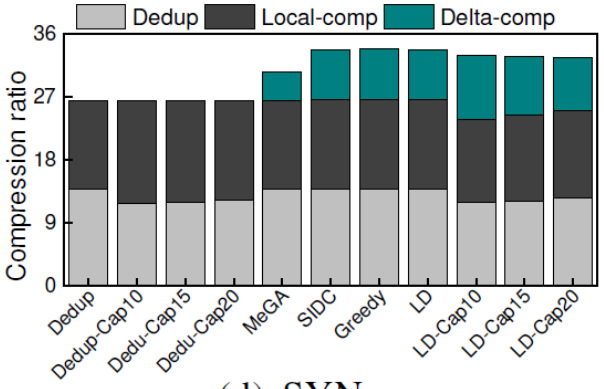
(a) RDB



(b) WEB



(c) CHRO

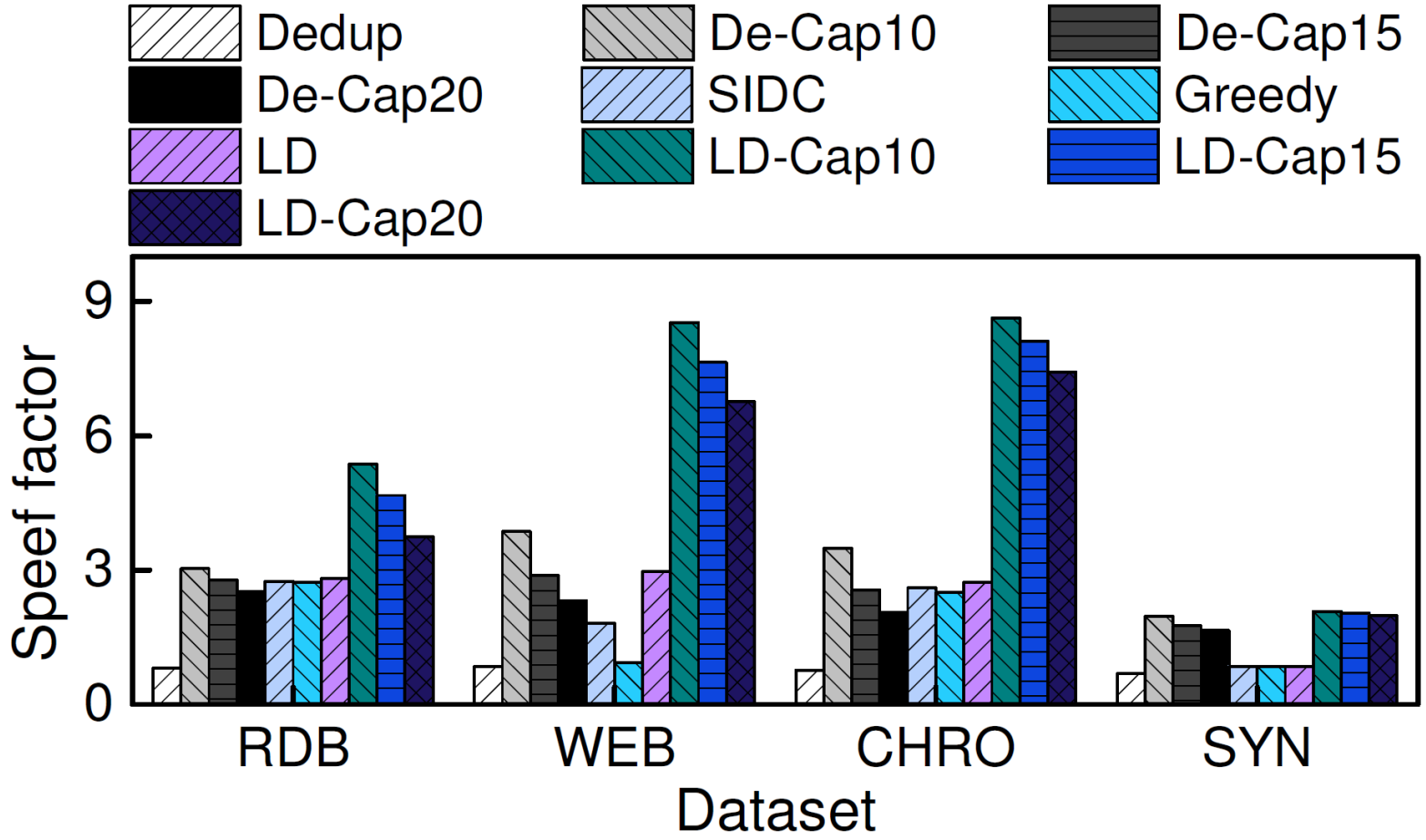


(d) SYN

LoopDelta在RDB、CHRO和SYN数据集上实现了与SIDC和Greedy相当的压缩比，并且高于 MeGA,同时在WEB数据集上实现了最高的压缩比。

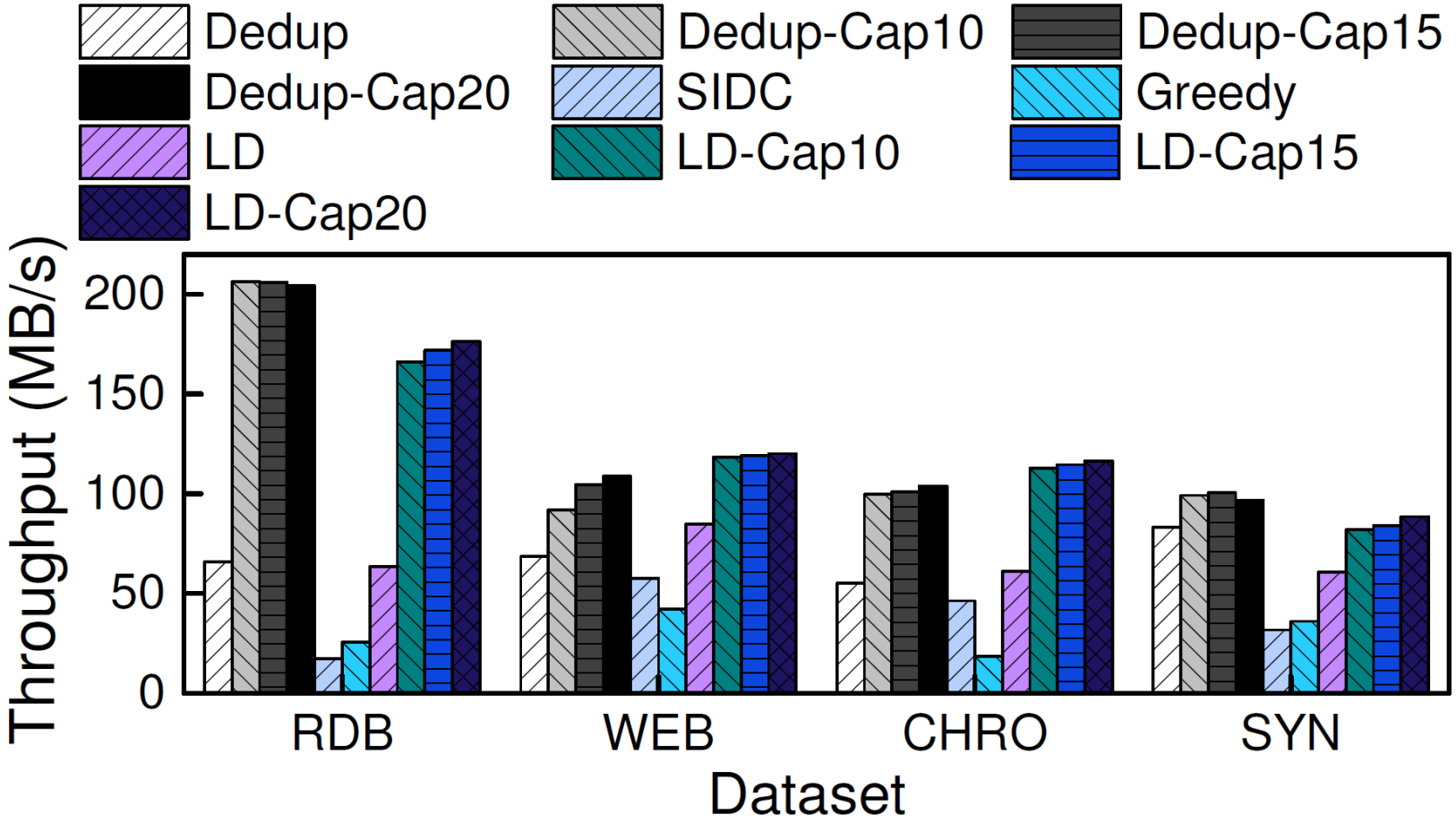
華中科技大學
计算机科学与技术学院
School of Computer Science &Technology，HUST

评估 6： 恢复性能



- LoopDelta > SIDC, Greedy
- LoopDelta-Cap > LoopDelta

评估 7： 备份吞吐量



- LoopDelta > SIDC, Greedy
- LoopDelta-Cap > LoopDelta


华中科技大学计算机科学与技术学院
School of Computer Science &Technology,HUST

汇报完毕
欢迎指正