Tair ldb(LevelDB)原理与应用案例

淘宝-核心系统部-存储组-那岩 (王玉法)

neveray@gmail.com

Tair 存储引擎概览

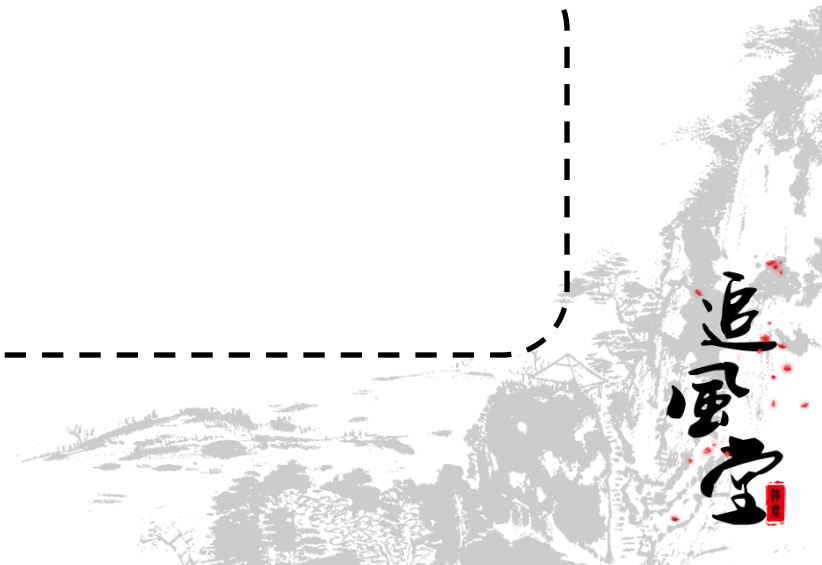
非持久化引擎

mdb
特性：kv以及分级key缓存

rdb (Redis)
特性：kv以及复杂数据结构缓存

持久化引擎

ldb (LevelDB)
特性：kv以及分级key存储，数据排序



目录

一．LevelDB原理

二．Tair ldb介绍

三．ldb应用案例

四．问题与后续计划

一．LevelDB原理

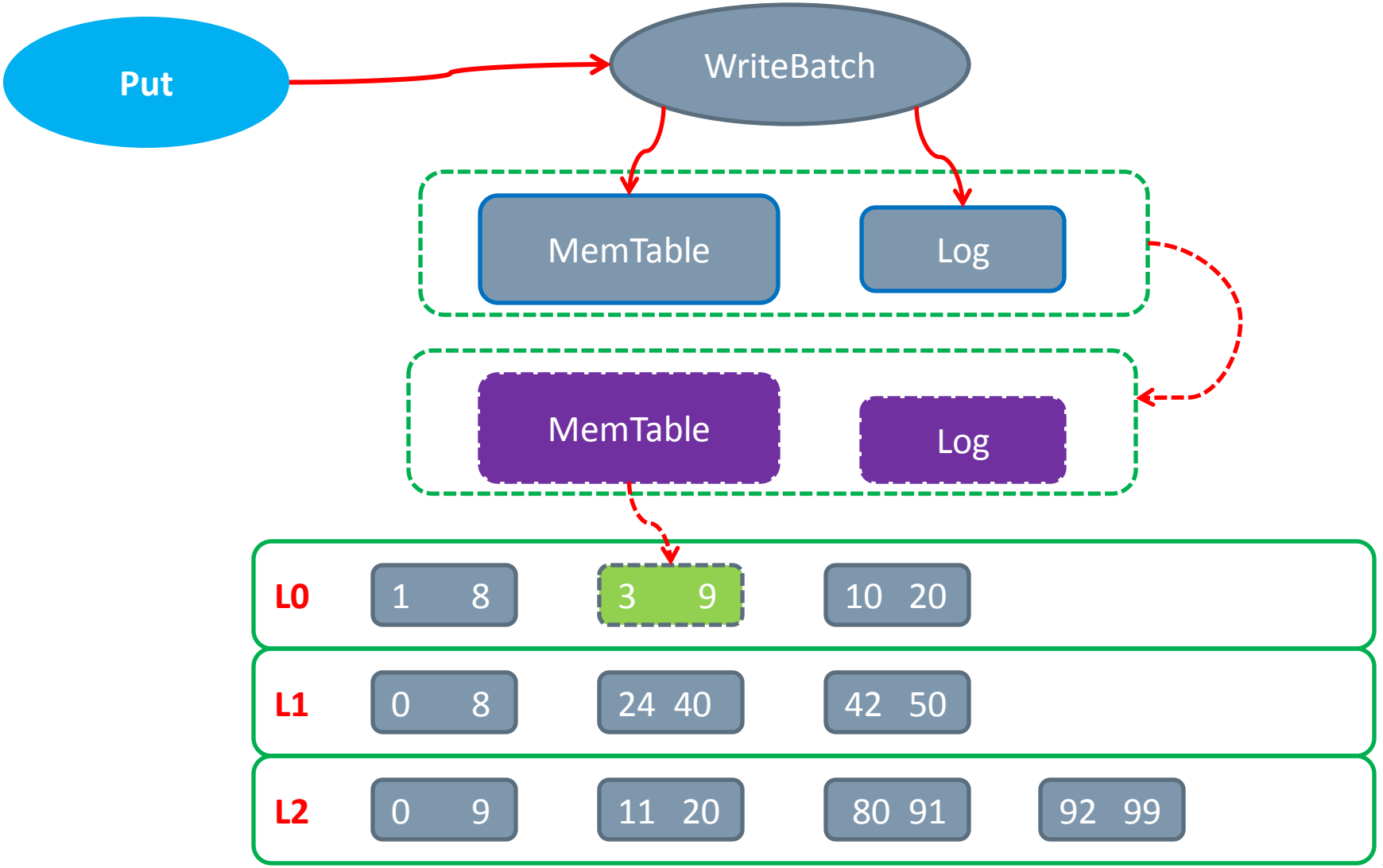
LevelDB概述

- Google开源的单机KV存储
- 内部排序，支持range遍历
- Merge-Dump模式
- 内存中排序MemTable

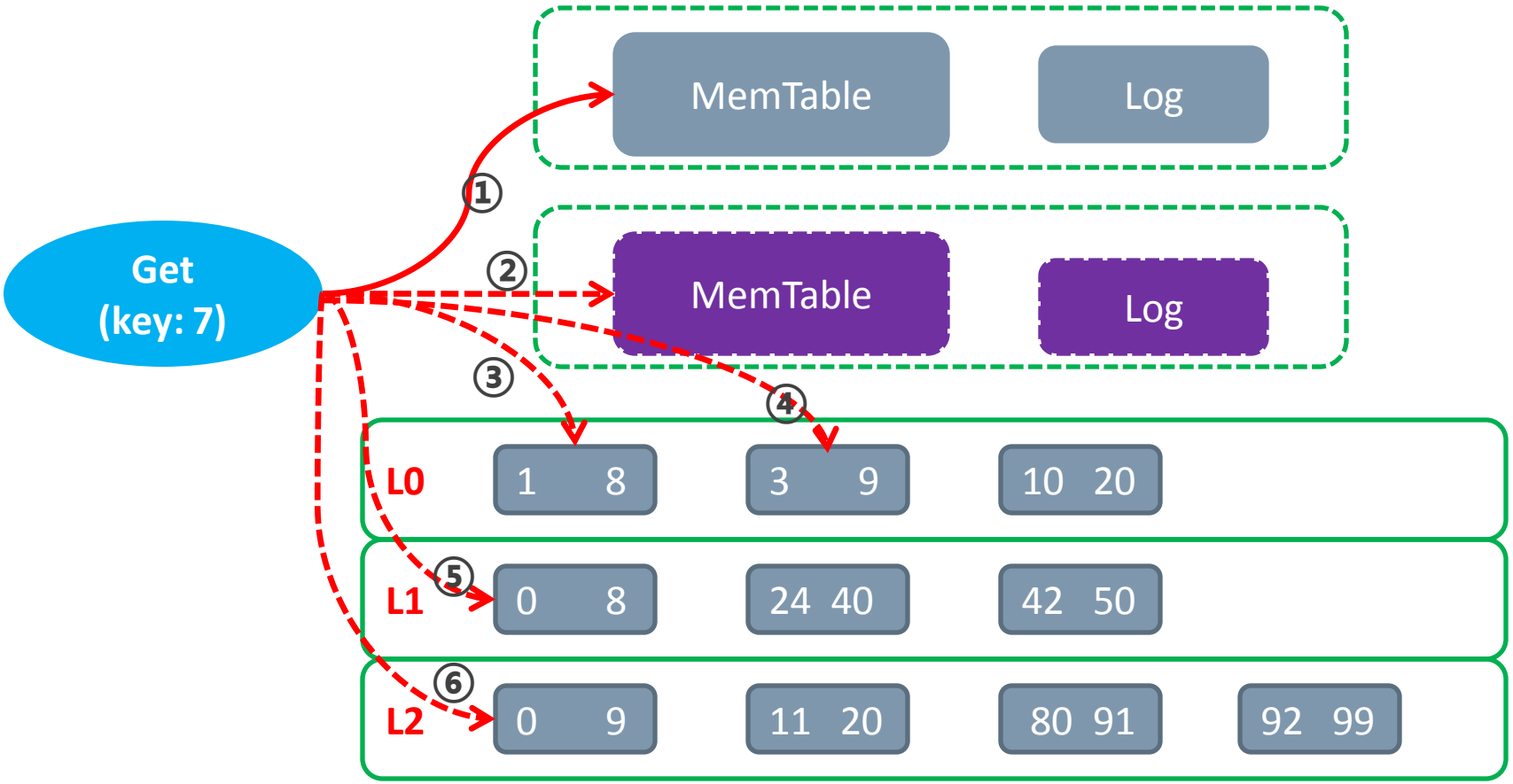
LevelDB概述

- 磁盘SSTable分level管理
- 后台Compact数据合并，level均衡
- 写为顺序IO，读为随机IO

LevelDB写流程


Put
WriteBatch
MemTable
Log
MemTable
Log
L0
1 8
3 9
10 20
L1
0 8
24 40
42 50
L2
0 9
11 20
80 91
92 99

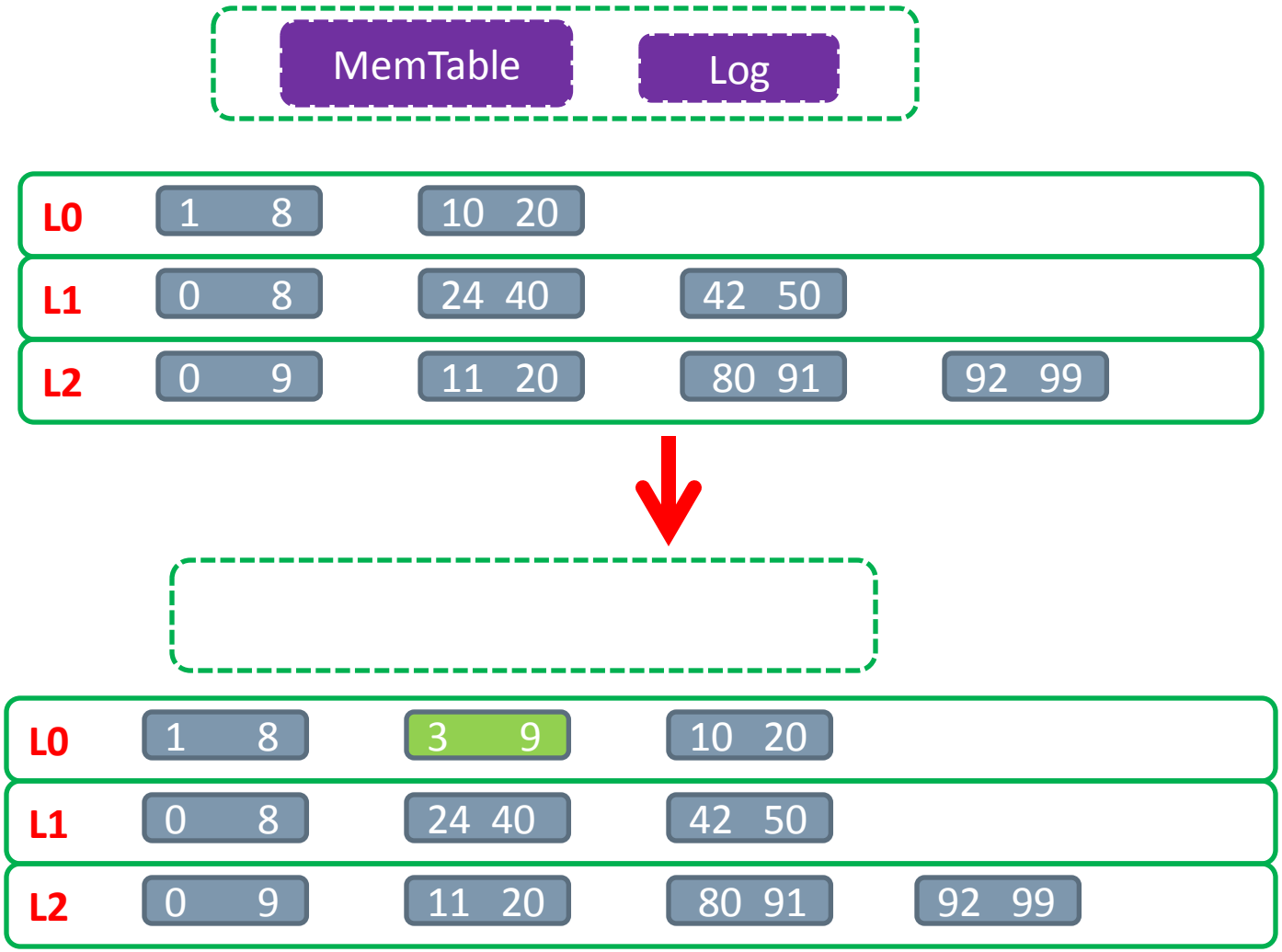
LevelDB读流程


MemTable
Log
MemTable
Log
Get
(key: 7)
①
②
③
④
⑤
⑥
L0
1 8
3 9
10 20
L1
0 8
24 40
42 50
L2
0 9
11 20
80 91
92 99

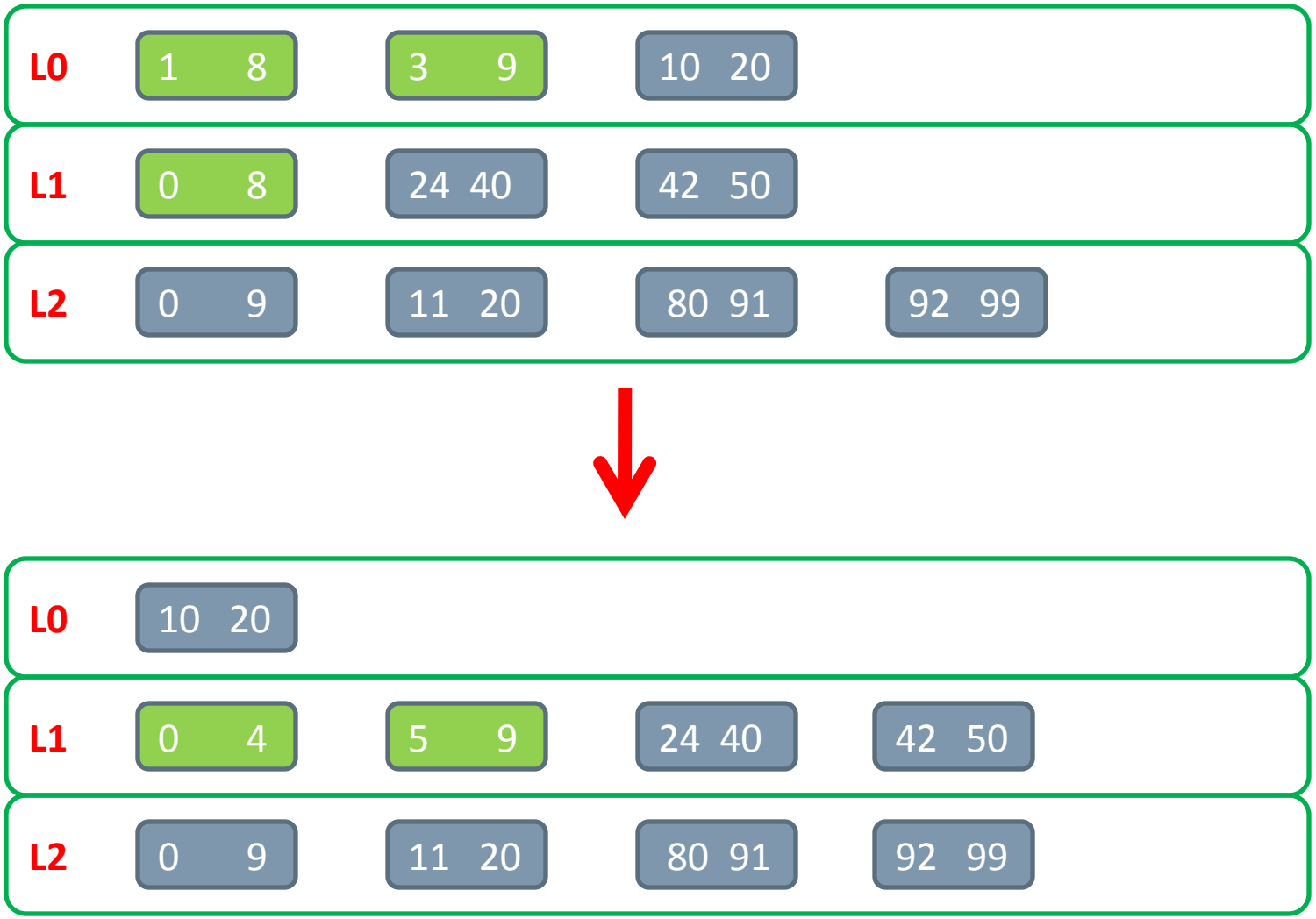
LevelDB Compact

- **memtable写满，直接dump成level 0上的sstable，无数据合并**
- **level 的compact权值**
  - level 0： sstable的个数
  - level 1-n: level上的sstable文件总大小
- **level循环compact range**

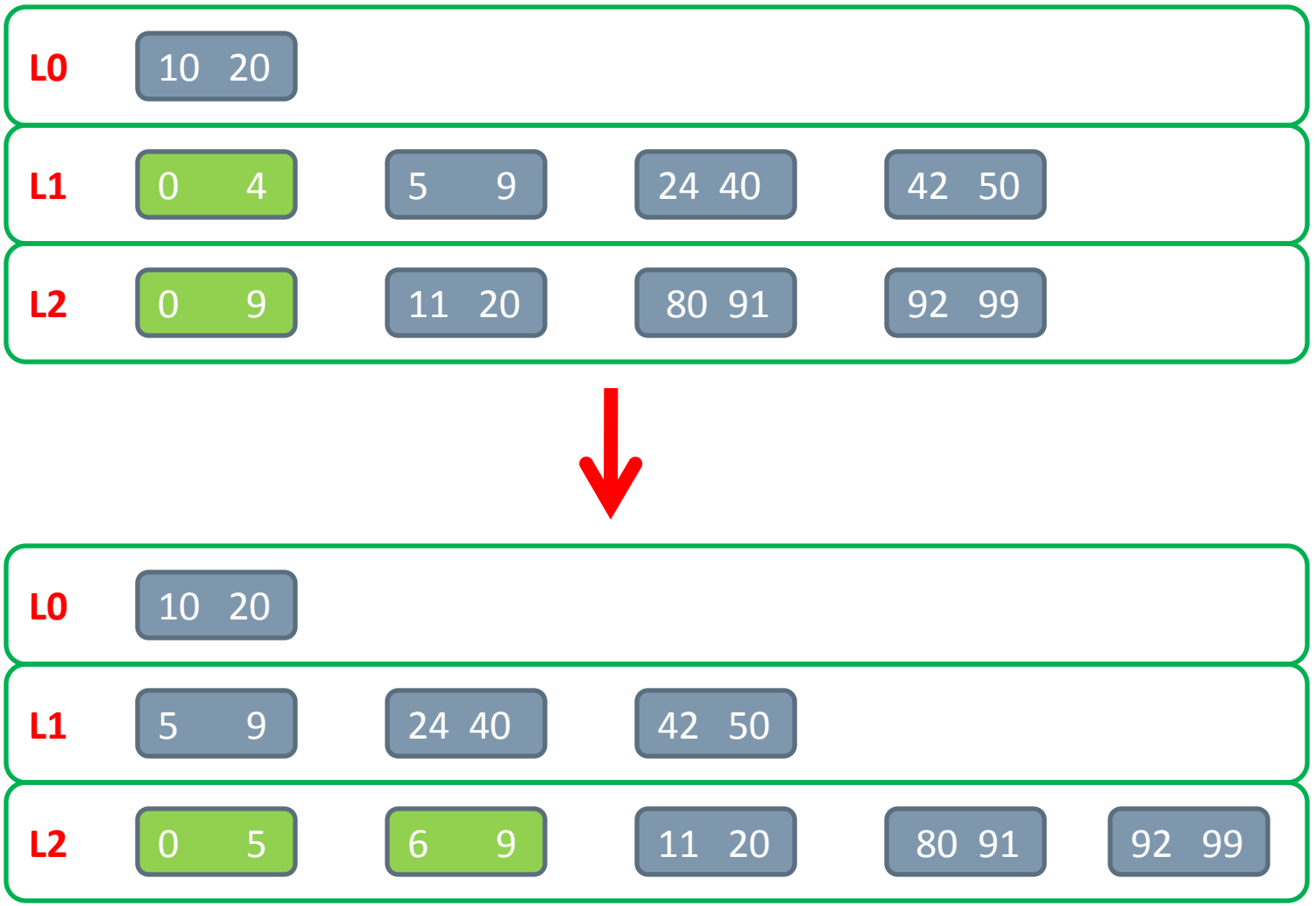
LevelDB Compact (Memtable/L0)



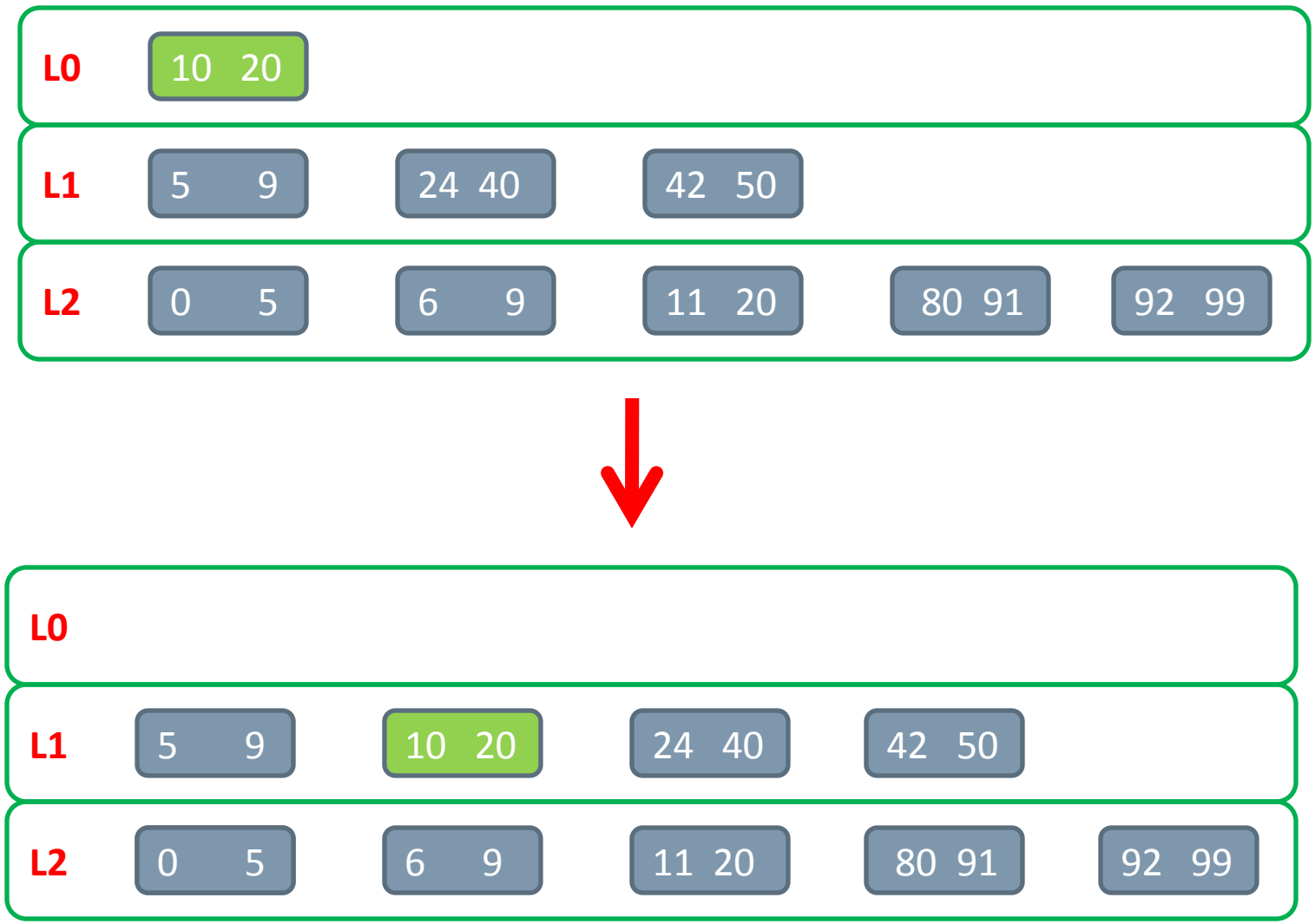
LevelDB Compact (L0/L1)



LevelDB Compact(L1/L2)



LevelDB Compact (L0/L1 Move)

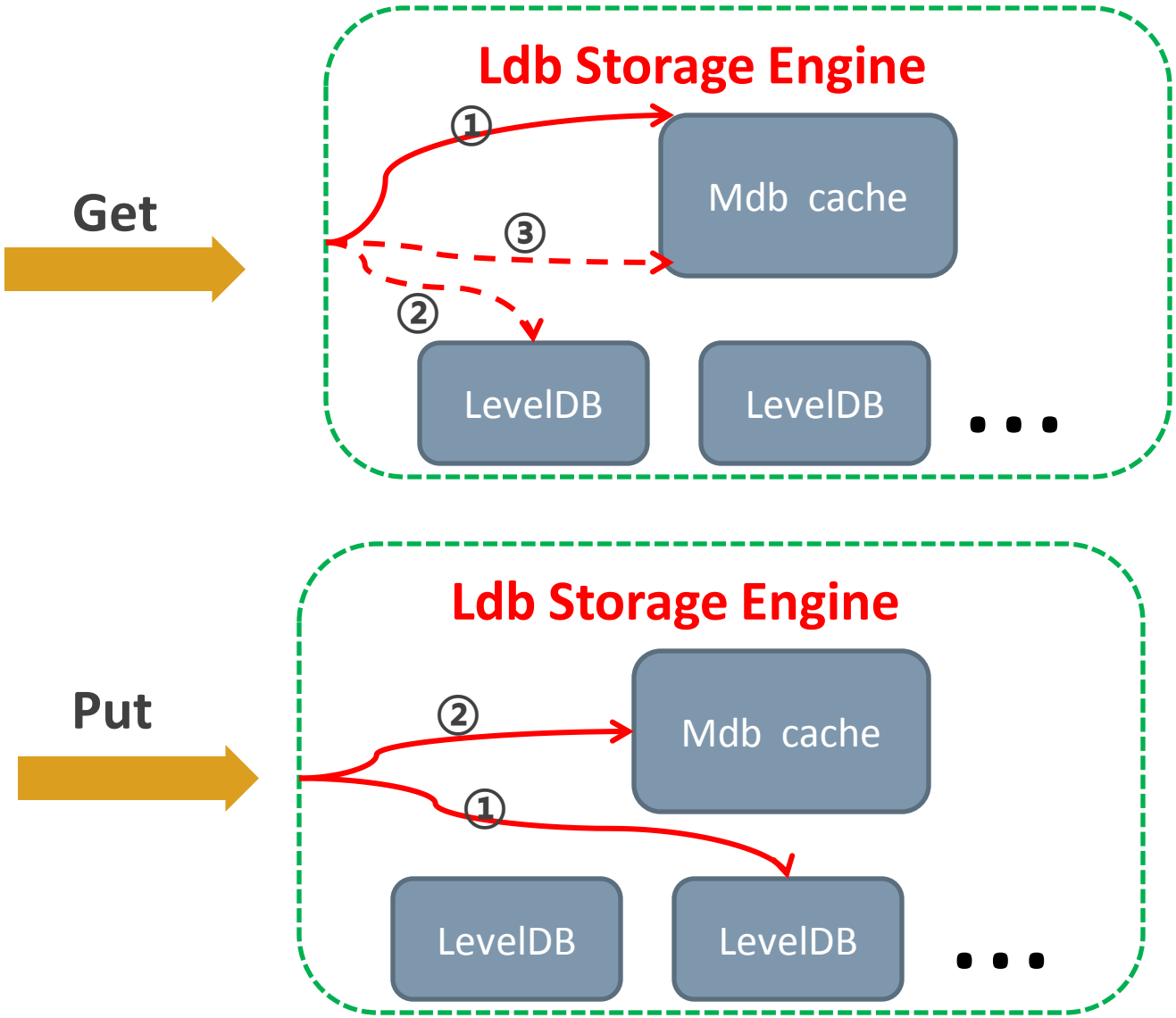


二．Tair ldb介绍

ldb概述

- 作为Tair的持久化存储引擎
- Tair的数据按桶管理
- 多实例配置，充分利用IO
- 数据过期/迁移，空间回收
- 内嵌Mdb作为内存KV级别cache
- SSD单机5w qps

Tair ldb 流程


Get
Ldb Storage Engine
①
Mdb cache
③
②
LevelDB
LevelDB
...
Put
Ldb Storage Engine
②
Mdb cache
①
LevelDB
LevelDB
...

三．ldb应用案例

ldb应用概述

- **counter持久化集群 ，访问量大，使用SSD**
  - 公用集群，通用排序方式
- **TC交易快照 ，数据量大， 使用SAS**
  - 数据key有特殊规律，采用特殊的排序方式，减少compact
- **数据定期做全量更新 ，广告bt应用（ FastDump ）**

FastDump场景

- 数据定期做全量更新
- 大数据量的更新效率
- 更新数据时提供正常服务
- 数据分area导入
- 平时只读不写（少量写）

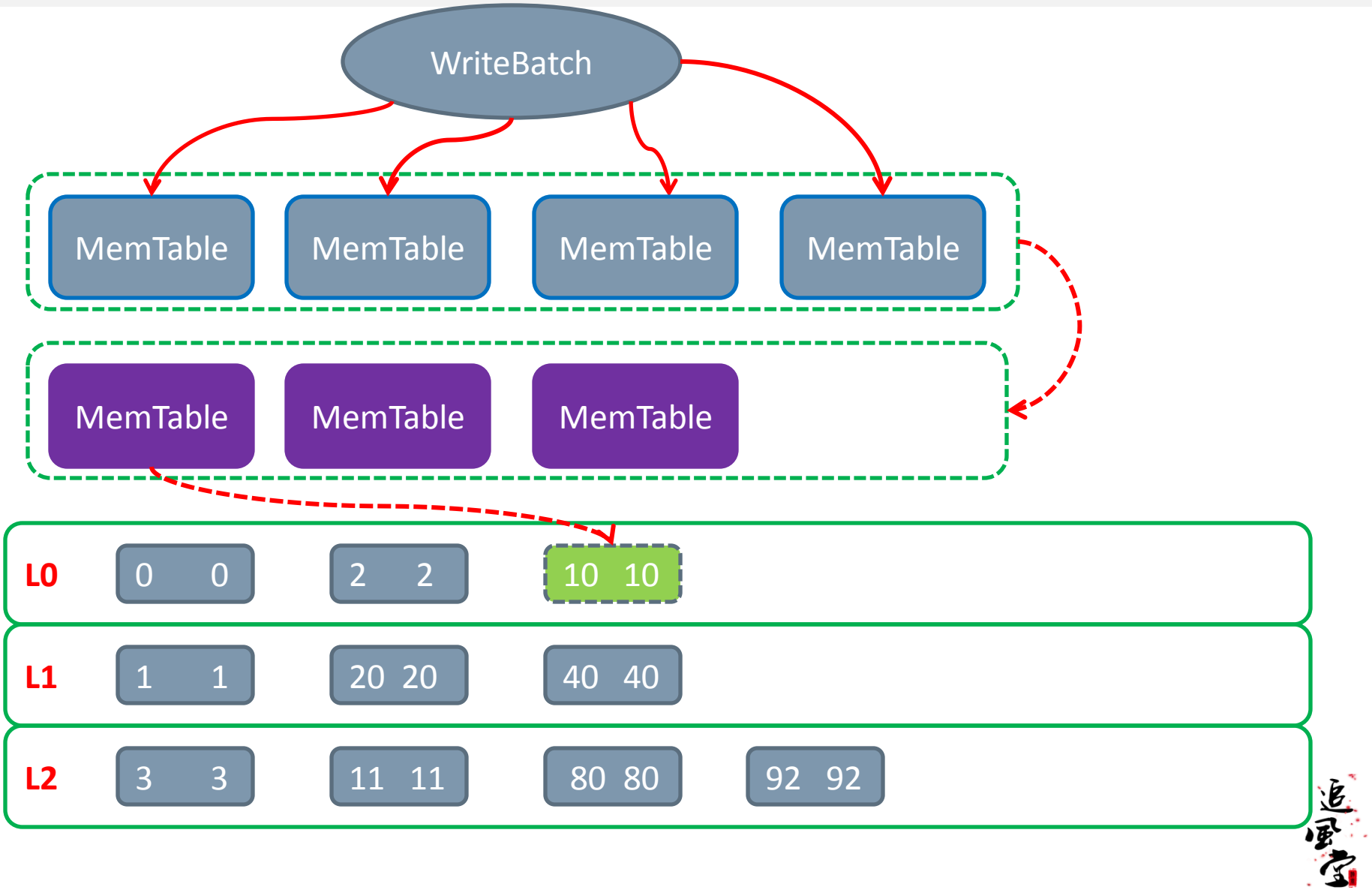
FastDump优化

- **全量更新，触发LevelDB内部大量compact：**
  - 根据ldb特点，数据按照桶预先划分排序
- **多桶数据并发插入：**
  - 均衡server端写入压力
  - LevelDB内部按桶划分MemTable
- **网络限制：**
  - 批量插入/通信包压缩

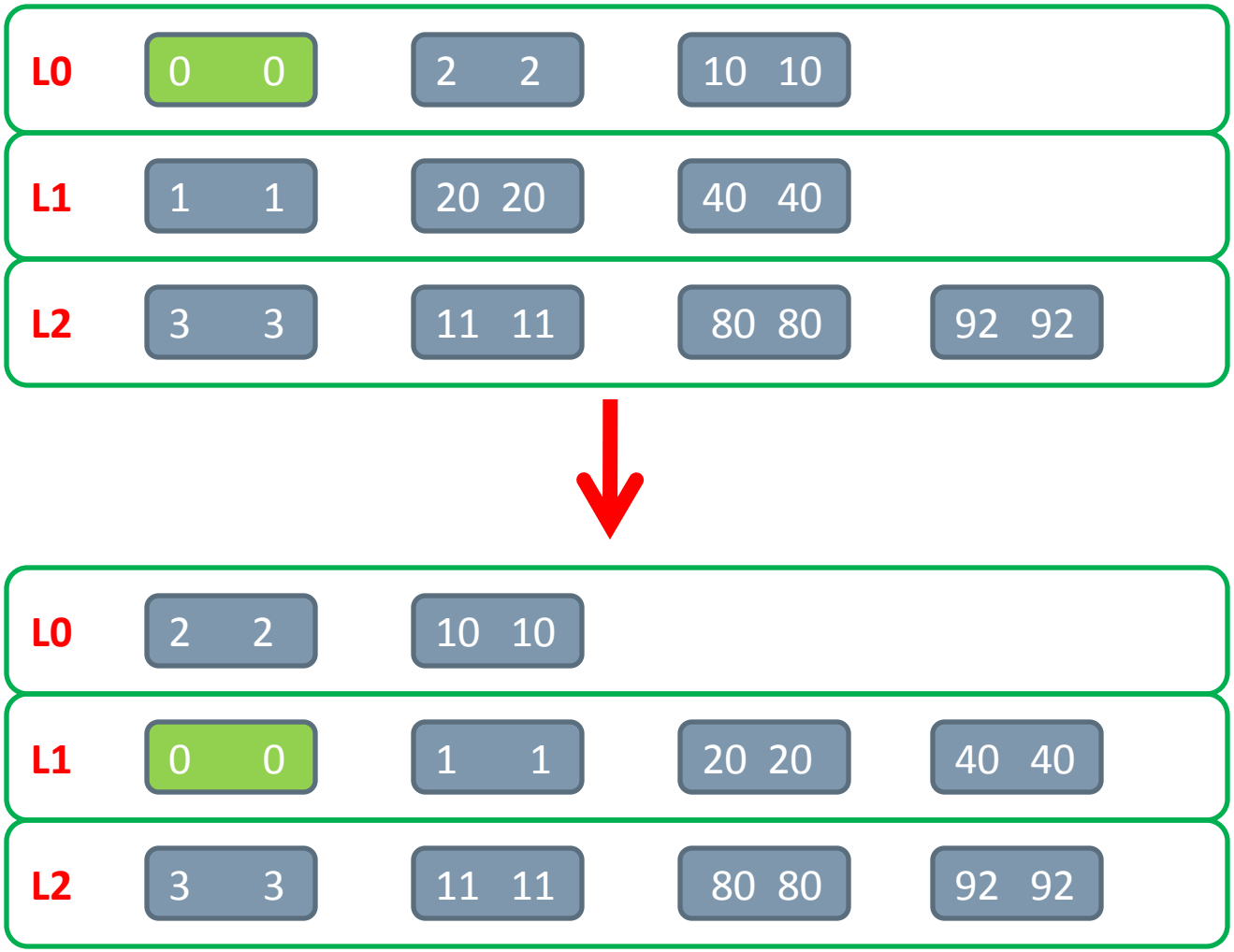
FastDump优化

- **LevelDB并发锁**
  - 合并批量WriteBatch
  - group commit
- **单机dump qps(KV size < 100byte)： 150w**

FastDump更新数据



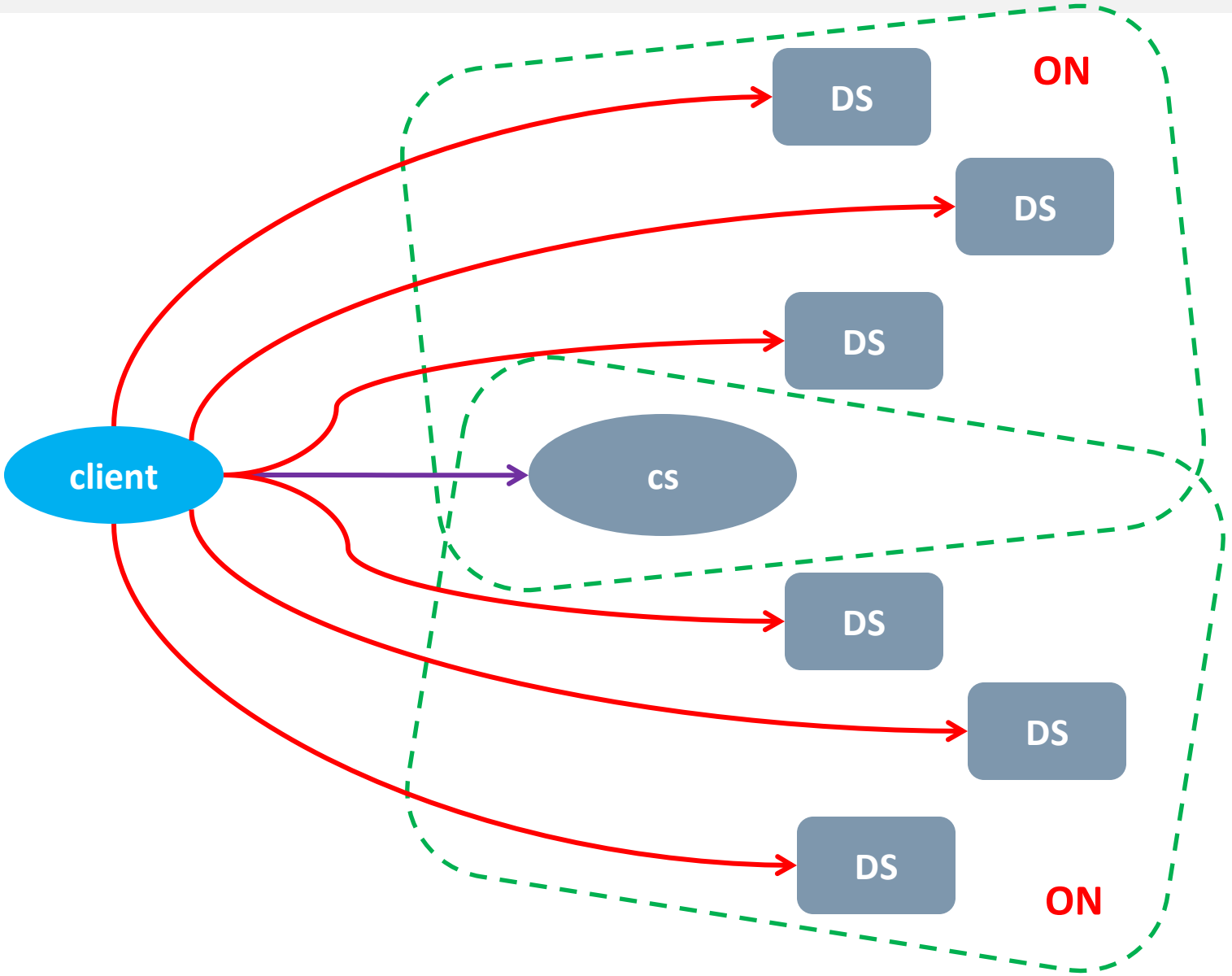
FastDump (Compact Move)



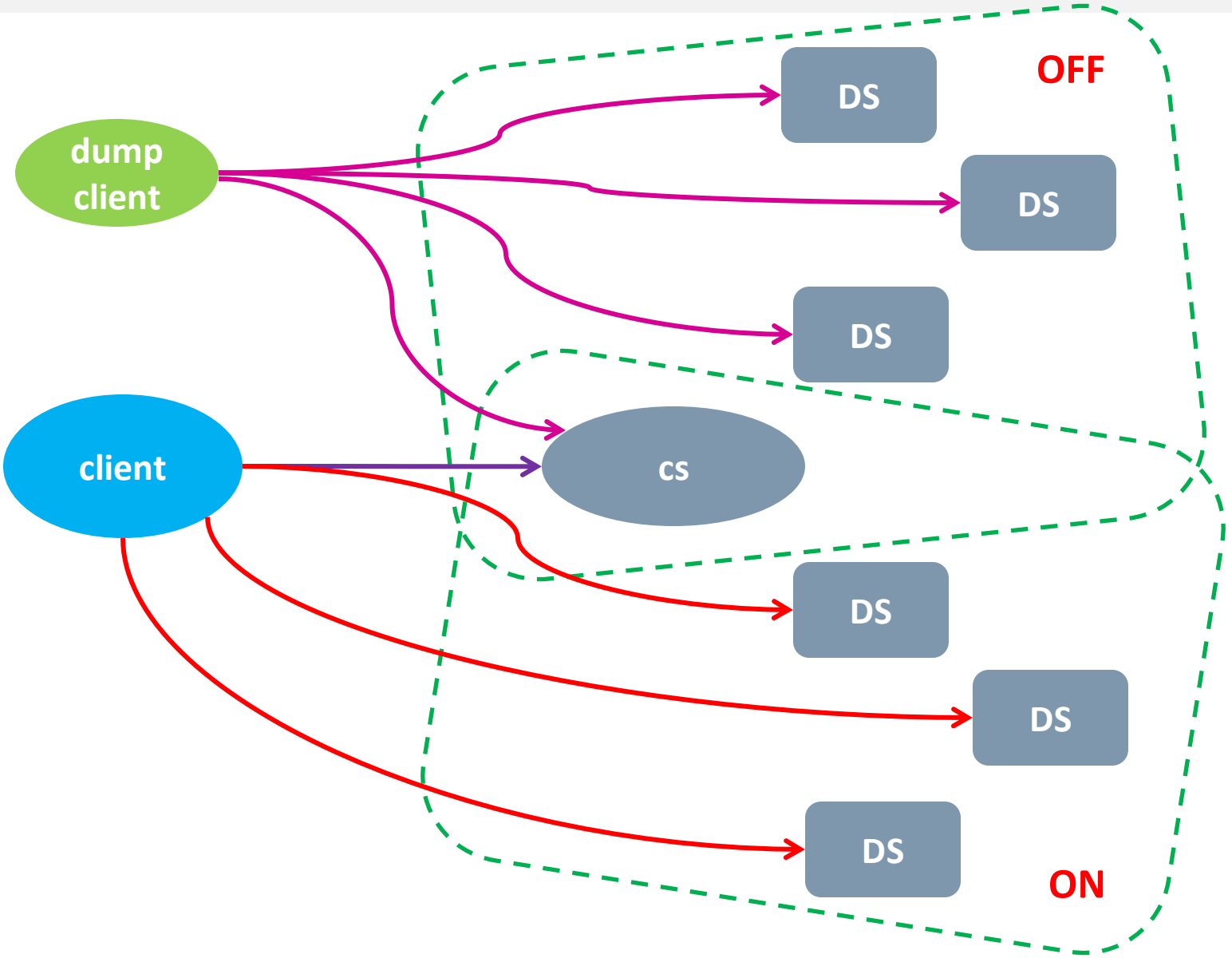
FastDump集群部署

- **平滑更新DB**
  - 多cluster部署方式
- **集群容灾**
  - client cluster/DS 粒度，透明切换访问

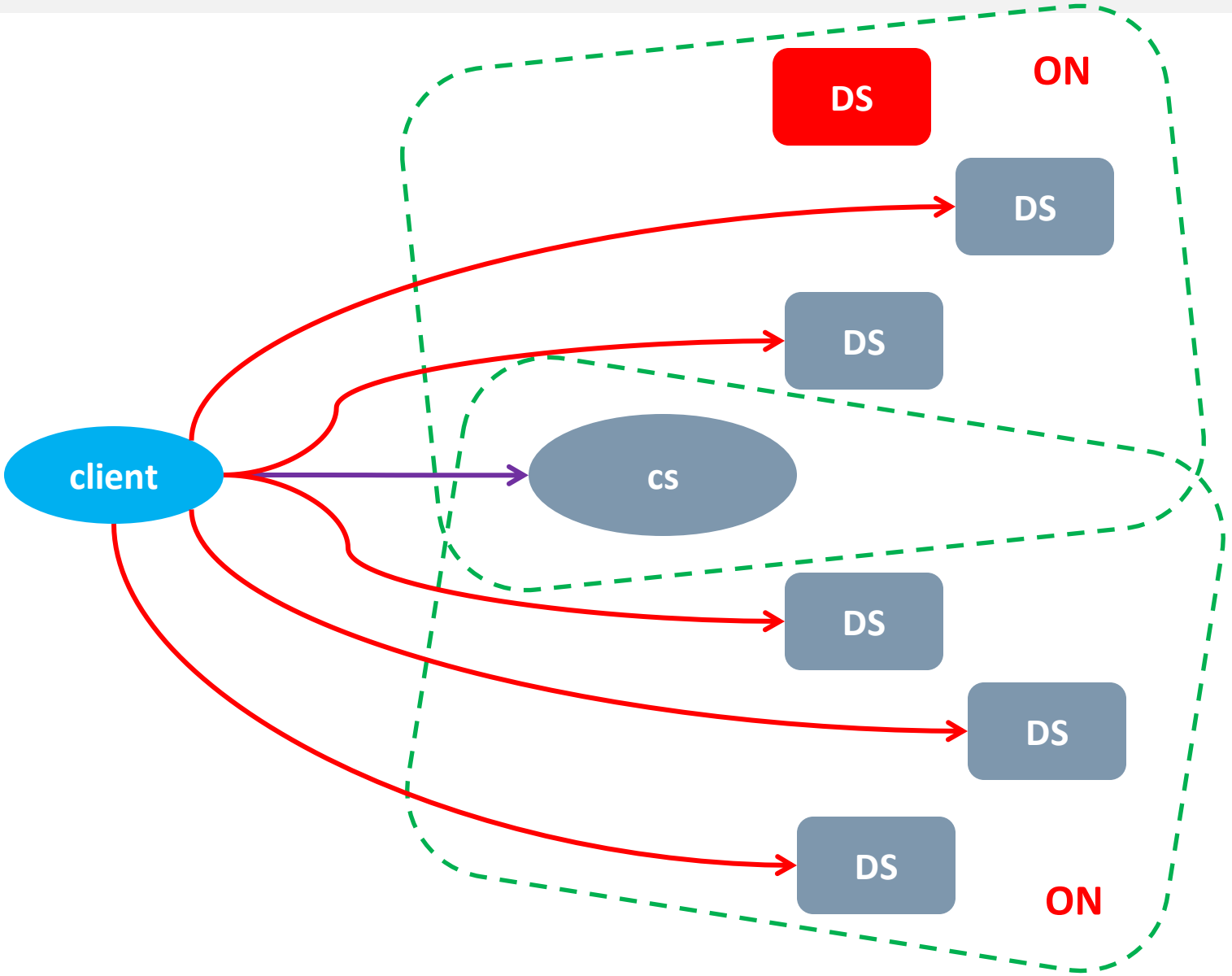
FastDump集群部署



FastDump(更新全量数据)



FastDump(DS down机)



四．问题与后续计划

问题与后续计划

- **机房容灾**
  - 双机房部署，应用读写切换
- **集群变化，数据迁移的速率低**
  - 数据按块迁移
- **网络先于磁盘(ssd)达到瓶颈**
  - 优化网络框架，更充分利用磁盘IO
- **访问不存在数据，数据稀疏range**
  - bloomfilter功能

问题与后续计划

- **数据随机写入，尤其更新量大时，compact造成的IO过多，磁盘IO量与实际写入数据量比例过大**
  - 内部range分布分析，策略降低高level的compact速率
- **FastDump切换db的粒度过大**
  - 支持集群内area粒度切换

相关资料

- **官方开源页面**

  http://code.google.com/p/leveldb/

- **LevelDB实现原理解析**

  http://rdc.taobao.com/blog/cs/?p=1378

- **Tair ldb实现介绍**

  http://rdc.taobao.com/blog/cs/?p=1394

谢 谢 ！