



2014中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2014



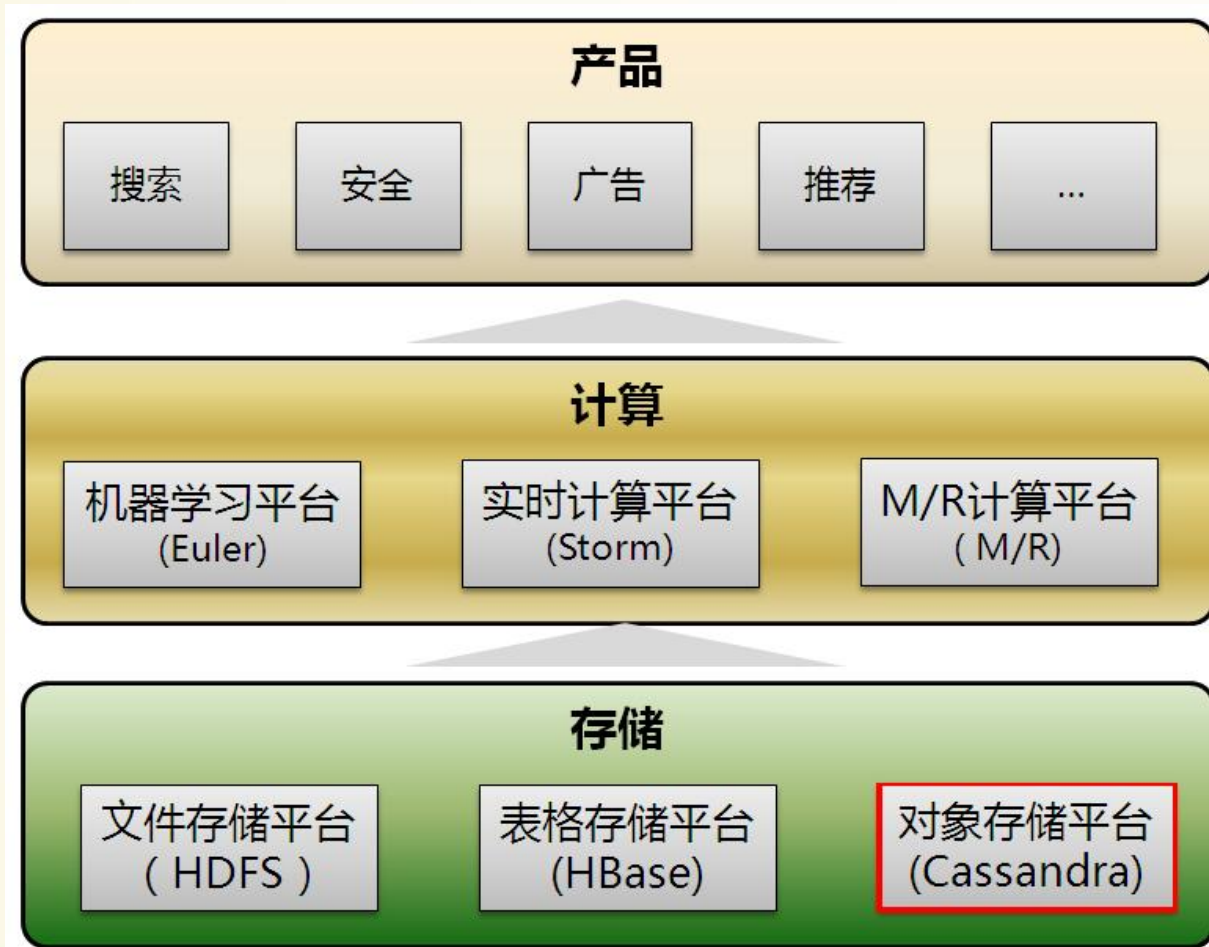
大数据技术探索和价值发现

Cassandra应用与改进

王 锋 @360



360的大数据





Cassandra

- 集群现状
- 问题与改进
- 后续工作

Cassandra集群现状

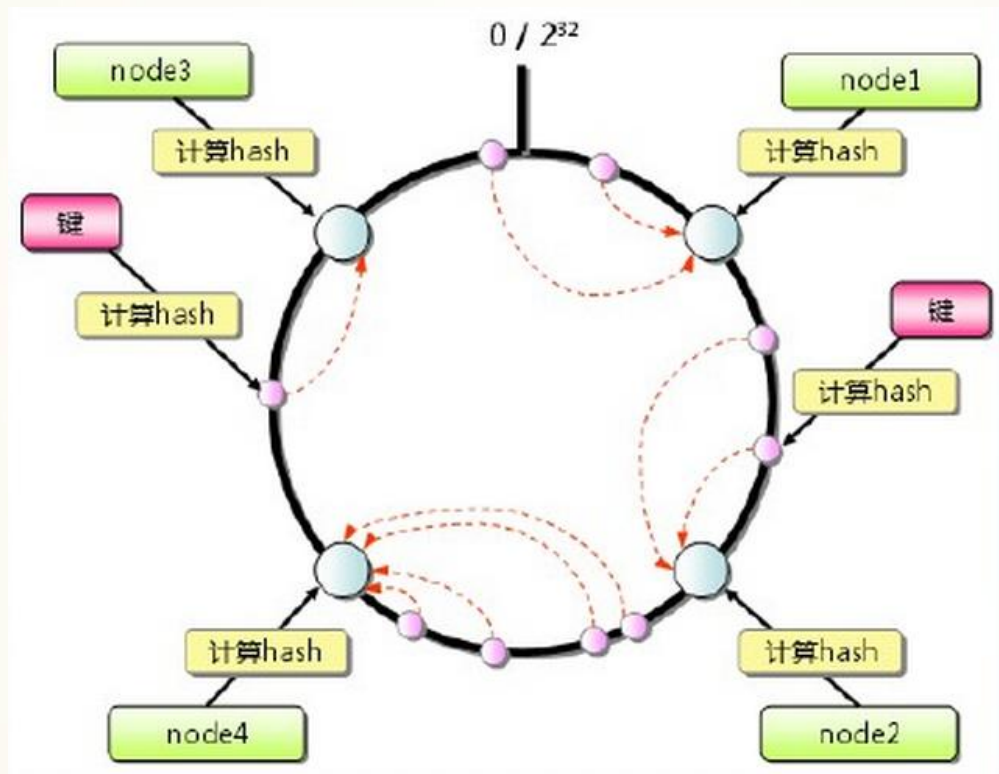
- 存储规模

主机规模	备份规模	数据规模	单日新增
8000台	3800台	70PB	400TB

单集群规模	集群容量
150台（24/3T）	9P
300台（24/3T）	18P
.....
1000台（24/4T）	84P

Cassandra特点

- Dynamo + BigTable
- 高可用性
- 可扩展性



Cassandra应用方式

- 使用方式

- 预先划分range:

- RandomPartitioner

- OrderPreservingPartitioner

- 文件I/O方式:

- standard 
 - mmaped

Address	Status	State	Load	Owns	Token
10.115.84.71	Up	Normal	13.71 TB	5.56%	170141183460469231731687303715884105726
10.115.84.72	Up	Normal	13.69 TB	5.56%	9452287970026068429538183539771339207
10.115.84.73	Up	Normal	13.71 TB	5.56%	18904575940052136859076367079542678414
10.115.84.89	Up	Normal	13.74 TB	5.56%	28356863910078205288614550619314017621
10.115.84.75	Up	Normal	13.72 TB	5.56%	37809151880104273718152734159085356828
10.115.84.90	Up	Normal	13.68 TB	5.56%	47261439850130342147690917698856696035
10.115.84.77	Up	Normal	13.71 TB	5.56%	56713727820156410577229101238628035242
10.115.84.78	Up	Normal	13.71 TB	5.56%	66166015790182479006767284778399374449
10.115.84.79	Up	Normal	13.73 TB	5.56%	75618303760208547436305468318170713656
10.115.84.80	Up	Normal	13.71 TB	5.56%	85070591730234615865843651857942052863
10.115.84.81	Up	Normal	13.51 TB	5.56%	94522879700260684295381835397713392070
10.115.84.82	Up	Normal	13.69 TB	5.56%	103975167670286752724920018937484731277
10.115.84.83	Up	Normal	13.71 TB	5.56%	113427455640312821154458202477256070484
10.115.84.84	Up	Normal	13.7 TB	5.56%	122879743610338889583996386017027409691
10.115.84.85	Up	Normal	13.72 TB	5.56%	132332031580364958013534569556798748898
10.115.84.86	Up	Normal	13.7 TB	5.56%	141784319550391026443072753096570088105
10.115.84.87	Up	Normal	13.71 TB	5.56%	151236607520417094872610936636341427312
10.115.84.88	Up	Normal	13.69 TB	5.56%	160688895490443163302149120176112766519
10.115.84.88	Up	Normal	13.69 TB	5.56%	170141183460469231731687303715884105726

Address	Status	State	Load	Owns	Token
10.140.92.175	Up	Normal	57.5 GB	6.25%	ffffff
10.140.92.176	Up	Normal	57.5 GB	6.25%	10000000
10.140.92.177	Up	Normal	57.51 GB	6.25%	20000000
10.140.92.178	Up	Normal	57.51 GB	6.25%	30000000
10.140.92.179	Up	Normal	57.51 GB	6.25%	40000000
10.140.92.180	Up	Normal	57.5 GB	6.25%	50000000
10.140.92.181	Up	Normal	57.5 GB	6.25%	60000000
10.140.92.182	Up	Normal	57.5 GB	6.25%	70000000
10.140.92.183	Up	Normal	57.5 GB	6.25%	80000000
10.140.92.184	Up	Normal	57.5 GB	6.25%	90000000
10.140.92.185	Up	Normal	57.5 GB	6.25%	a0000000
10.140.92.186	Up	Normal	57.5 GB	6.25%	b0000000
10.140.92.187	Up	Normal	57.5 GB	6.25%	c0000000
10.140.92.188	Up	Normal	57.5 GB	6.25%	d0000000
10.140.92.189	Up	Normal	57.5 GB	6.25%	e0000000
10.140.92.191	Up	Normal	57.5 GB	6.25%	f0000000
10.140.92.191	Up	Normal	57.5 GB	6.25%	ffffff

改进的重心

- 数据可靠性
- 运维的便捷
- 成本的考量

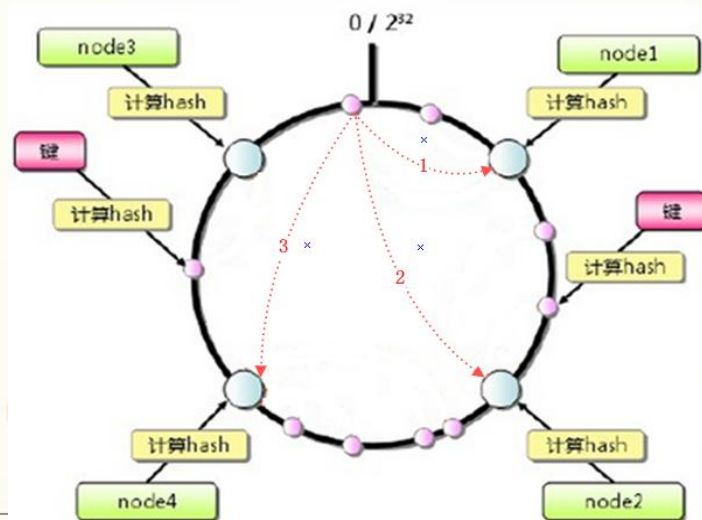
数据可靠性：本地自检修复

- 问题：

- 扇区、磁盘故障、主机故障导致副本缺失
- 新写入数据副本可能不足（ONE/QUORUM）
- 系统自带机制不能保证副本及时修复：
 - 读修复、Hinthandoff、Repair操作局限性
 - 损坏的SSTable在内存索引中，但磁盘数据读异常

- 改进：

- 故障磁盘/文件自动在线摘除
- 接入节点新增数据的副本检查
- 数据节点全量数据的扫描修复



数据可靠性：本地自检修复

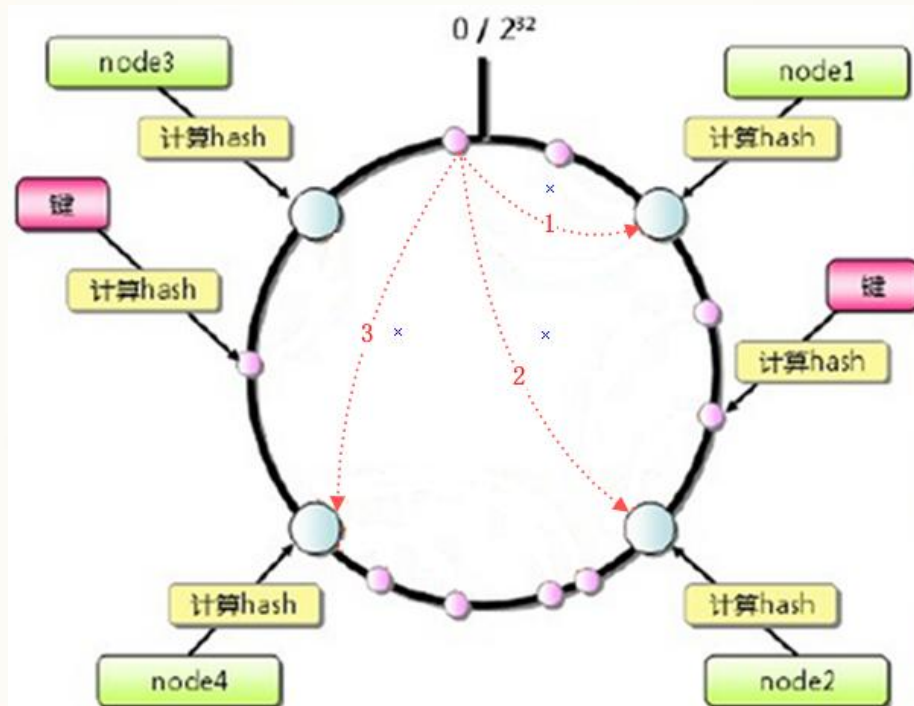
- 1. 文件/磁盘自动摘除

- 目的：

- 去腐生肌
 - 消除影响

- 基于统计

- 文件异常访问次数
 - 摘除文件比例



数据可靠性：本地自检修复

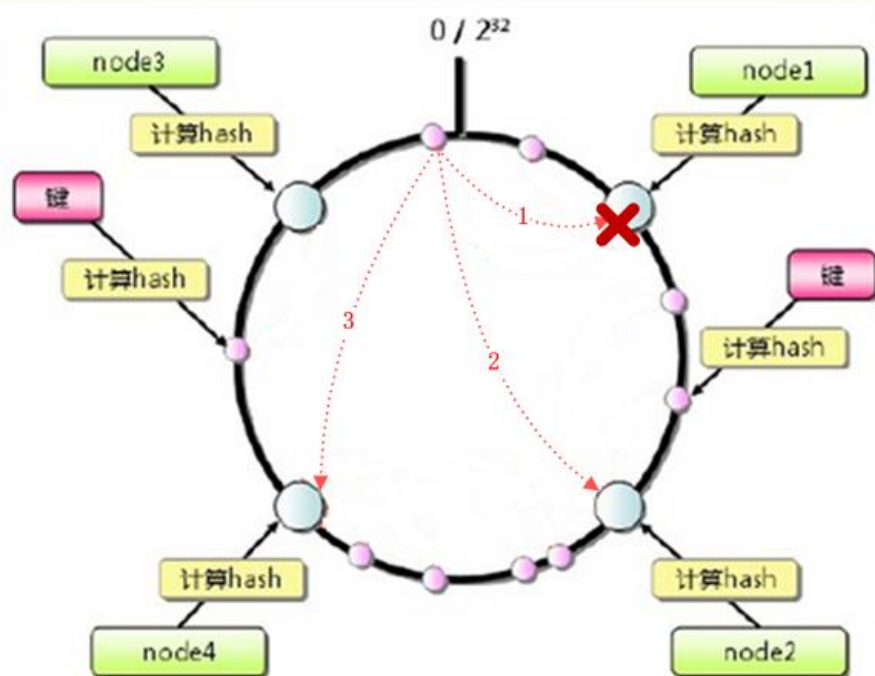
- 2. 数据节点：全量数据的扫描修复

- 目的：

- 磁盘/文件故障摘除即启动修复
- 尽快恢复全副本的状态

- 修复方式：

- 确定故障所属Range
- RowScan + Diff
- KeyScan + Read (ALL)



数据可靠性：本地自检修复

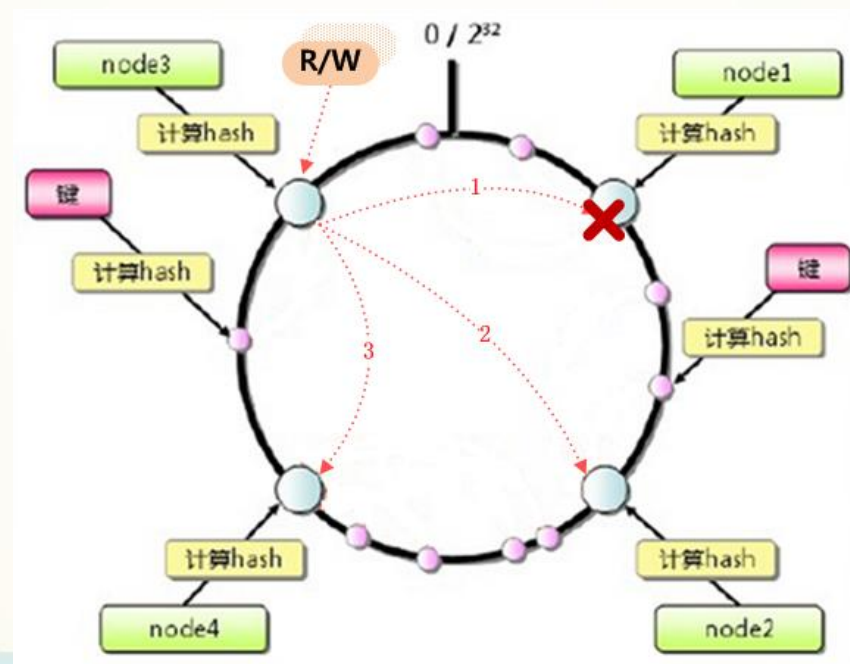
- 3. 接入节点：增量数据的检查修复

- 目的：

- 保证新写入数据副本数足够
 - 解决hinthandoff缺点

- 处理方式：

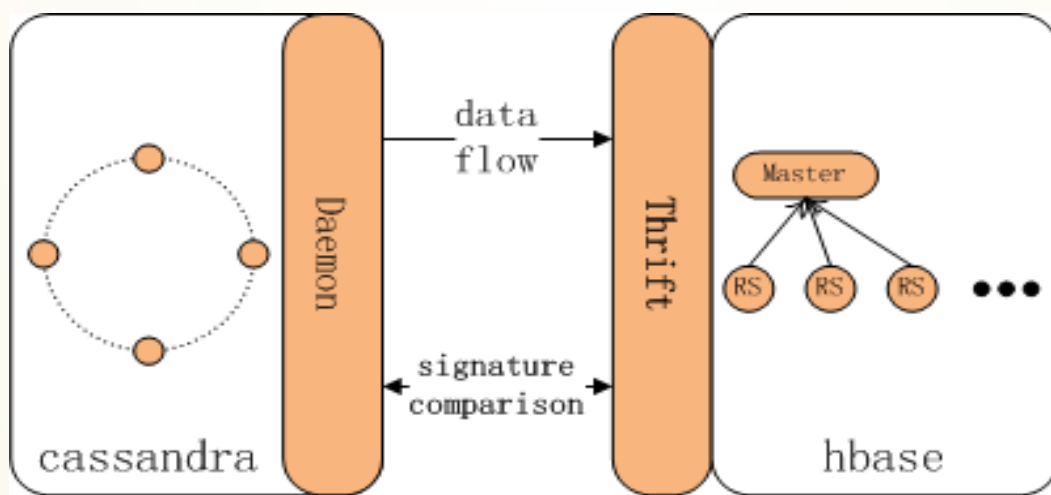
- 新增辅助表：proxycheck
 - 副本不足记入辅助表
 - 数据节点写失败：超时/拒绝
 - 数据节点停机
 - 读修复



数据可靠性：远程备份检查

- 问题：
 - 数据远程备份的必要性
 - 跨机房数据容灾
 - 跨机房流量切换
 - 消息队列，如kafka的局限
 - 大数据存储和性能方面
 - 增加了运维复杂度

- 改进：
 - 数据同步与检查机制
 - 辅助表做为缓冲队列
 - 节点主动式同步
 - 各节点负载均匀
 - 带宽占用易于控制
 - 运维简单方便



虚拟目录存储

- 问题：
 - 节点数据量大，SSTable文件多，磁盘空间导致无法做Major消重
 - SSTable文件数多，Scan操作导致CPU消耗严重
- 改进：
 - 将所管理range范围切分成若干子范围（目录）存储
 - 将每日新增文件通过Compaction分散到各目录中去
 - 各目录数据量小，可以分别做Major
 - 每次Scan请求只需打开某个虚拟目录的少量文件，CPU消耗降低

Compaction改进

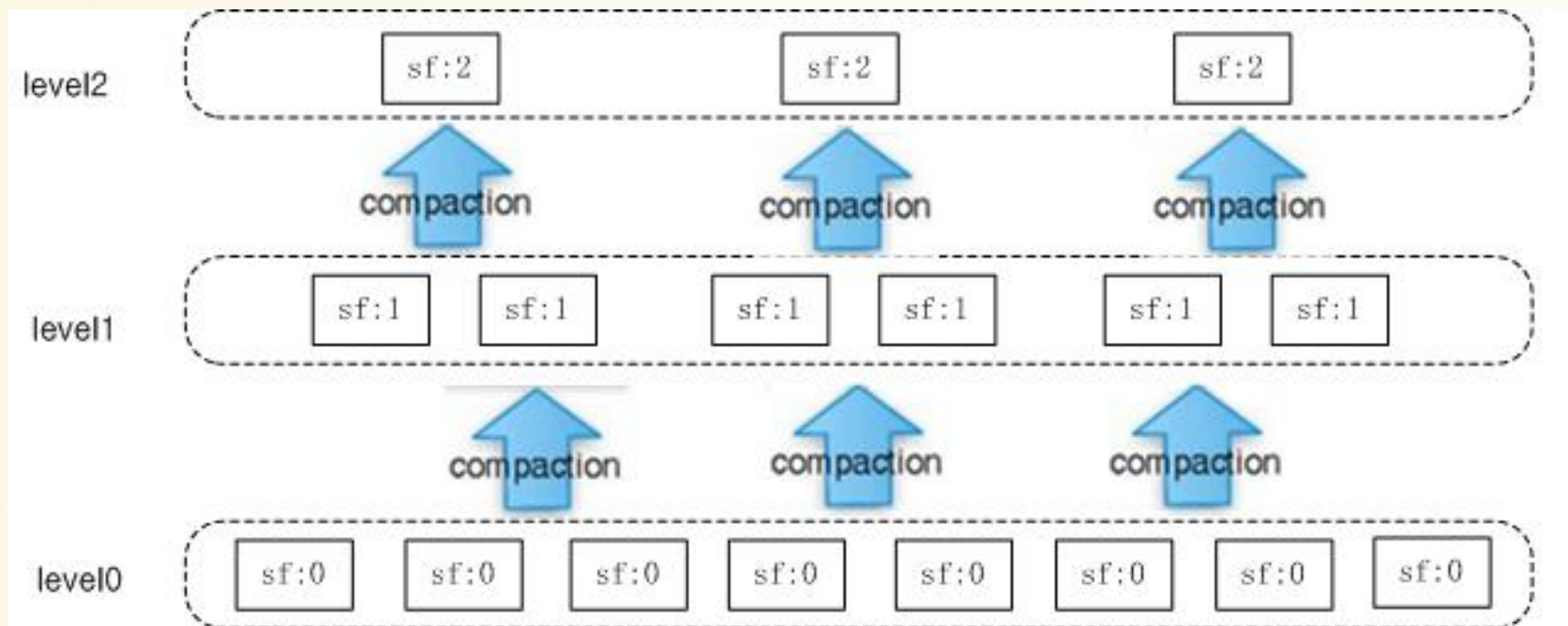
- 问题:

- 原有Compaction机制（SizeTiered/Leveled）较难避免数据重复参与Compaction的问题
- 尤其SizeTiered按文件大小分组Compaction，插入删除频繁的业务难以消重

- 改进:

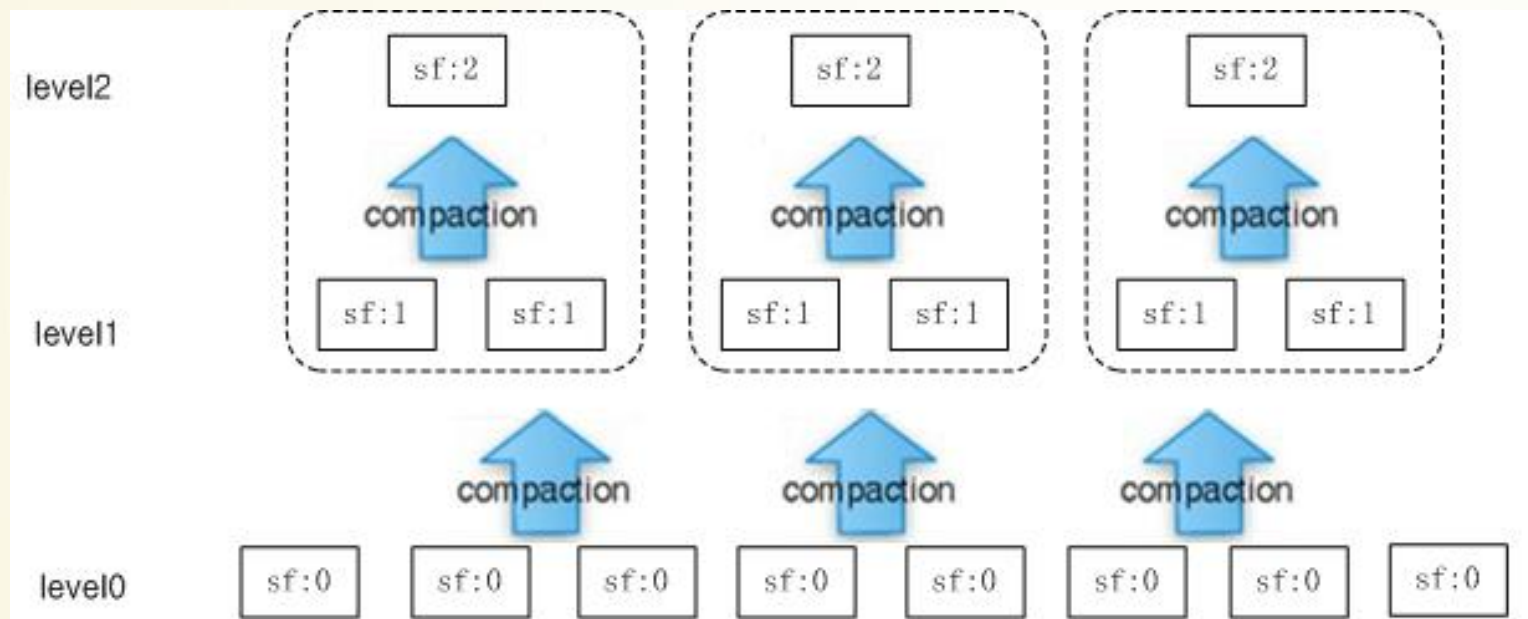
- SSTable增加level概念，标识做过Compaction的次数
- 新生成的文件level=0，每做一次level加1
- 每日新增文件筛选：时间 + level0

Compaction改进



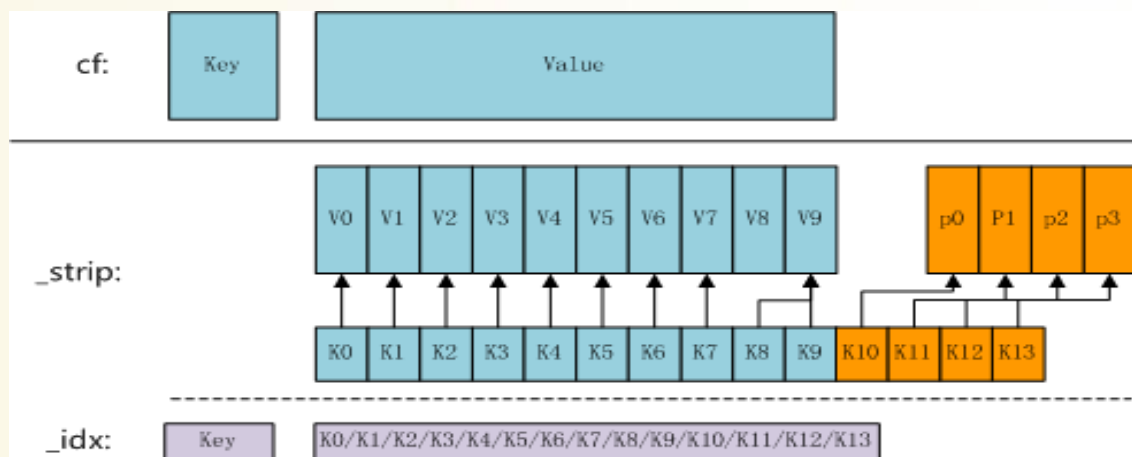
Compaction + 虚拟目录

- 配合虚拟目录存储：消重 + 减少重复I/O



EraserCode

- EraserCode方案：
 - 成本考虑：
 - 3副本 → 1.4副本 (10+4)
 - 数据可用性考虑：
 - 副本方式：连续3台机器磁盘故障，数据必丢失
 - 条带方式：相邻14台机器故障任意4台仍可修复



EraserCode




- 问题:

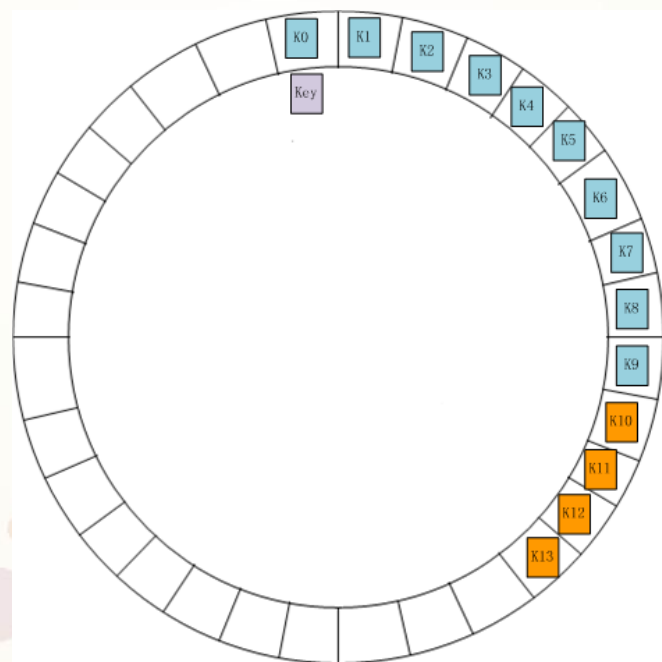
- 条带数据散布



- 数据均匀分布
 - 保证数据可靠性
 - 重复数据可消重

- 方案:

-  元数据保持多副本方式
 -  ~  按环顺序分布
 - 编码方案在线可调
 - 指令级加速



后续工作

- 内容：
 - Keyspace级别的环结构(空间划分)
 - Commitlog并发写
 - 磁盘故障预测下线反馈
 - 桌面级盘的尝试
 - 多网卡支持

Q&A

THANKS

