

SDCC 2017 | 上海

数据库核心技术与应用实战峰会

CSDN

MongoDB 秒级备份恢复

个人简介

```
{  
  Name:      “张友东”,  
  Company:   “阿里云”,  
  Position:  “技术专家”  
  Focus:     “分布式存储与数据库”,  
  Contacts:  {  
    Emal:    “zyd_com@126.com”,  
    Blog:    “yq.aliyun.com/users/1134812”,  
    Weibo:   “@HUST张友东”,  
  }  
}
```

历史的经验教训



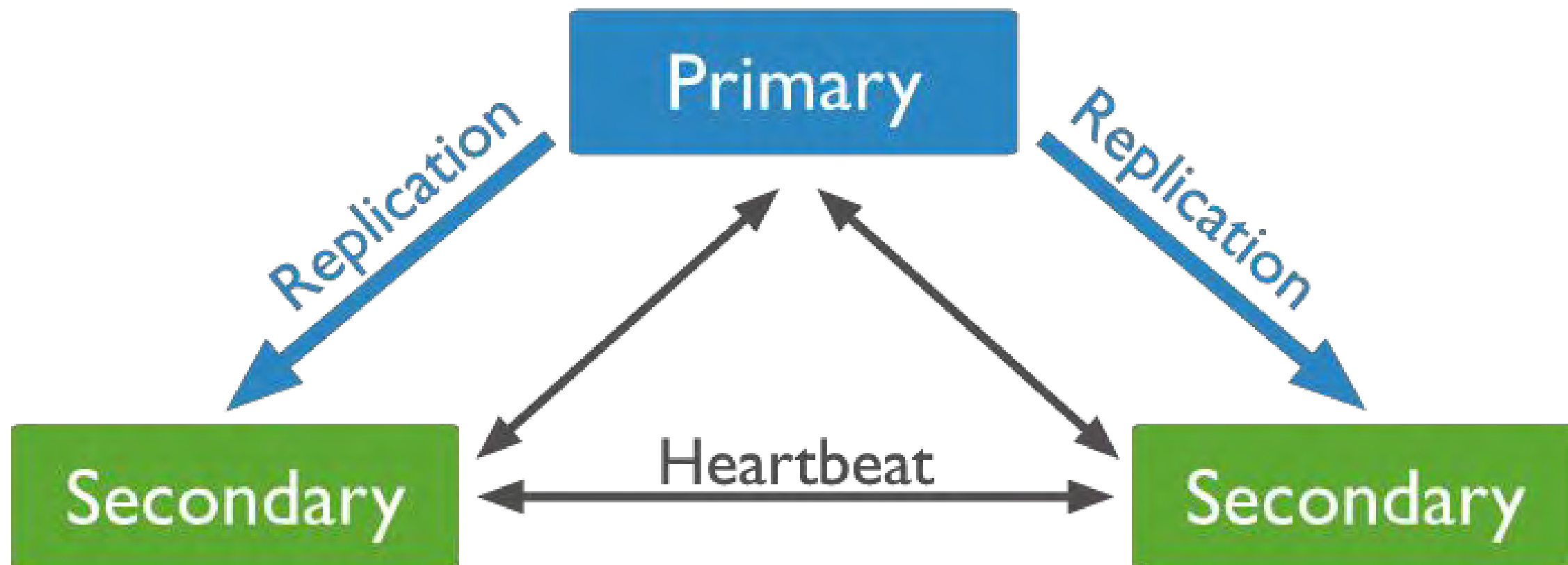
2017年1月
你的数据库备份了吗？



2017年2月
你的数据库备份有效吗？

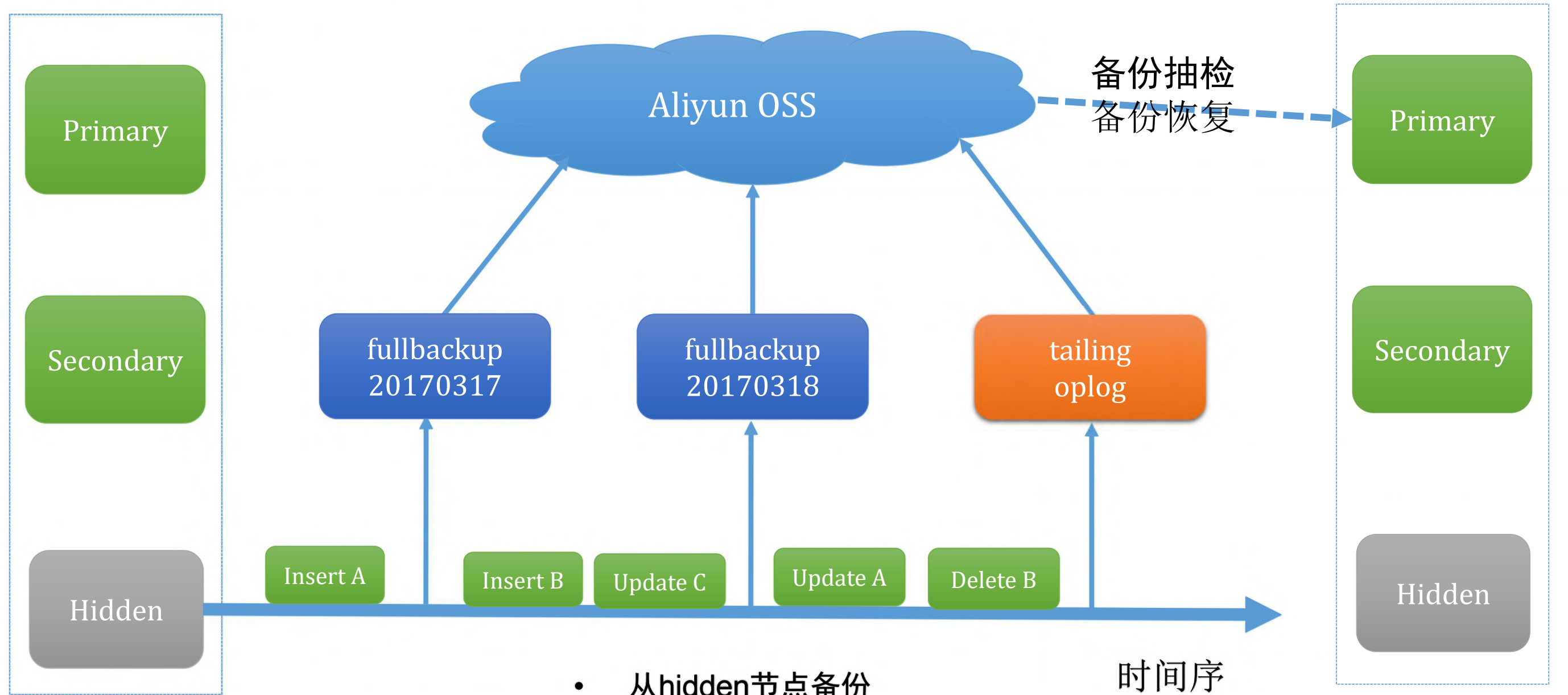
Next?

MongoDB复制集



思考：3节点复制集为什么还需要备份？

MongoDB云数据库备份/恢复



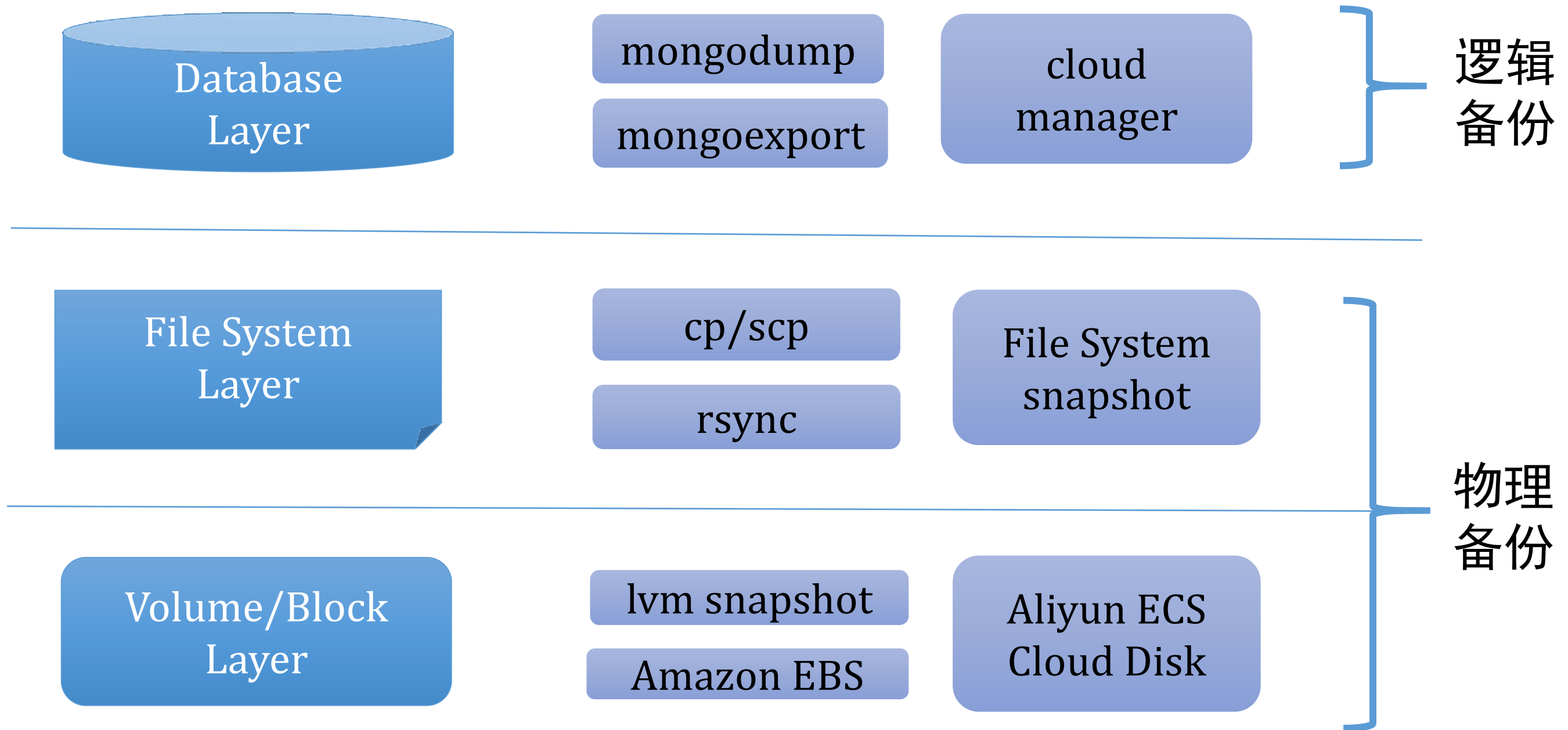
MongoDB
Replica Set

- 从hidden节点备份
- 每天一次全量备份
- 持续拉取oplog增量备份
- 定期巡检备份有效性
- 恢复时克隆到新实例

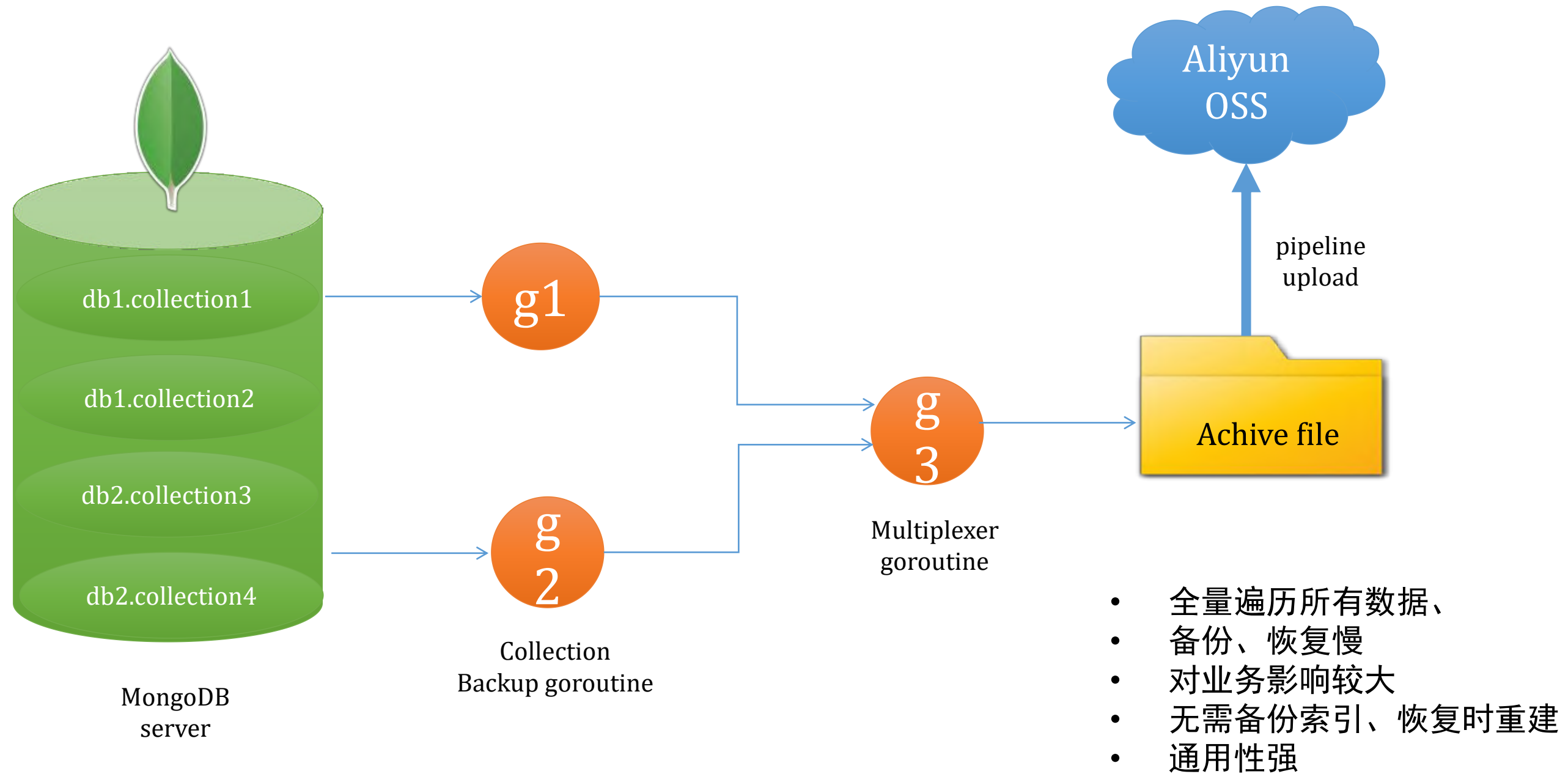
时间序

MongoDB
Replica Set

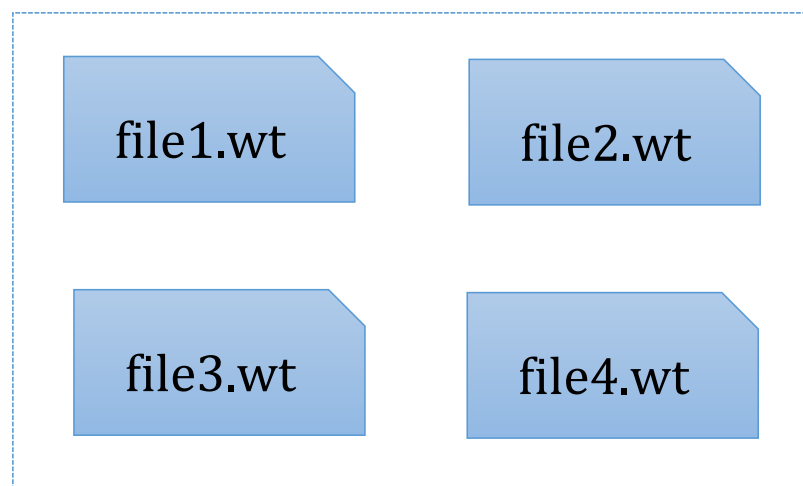
全量备份方法



逻辑备份流程 - mongodump

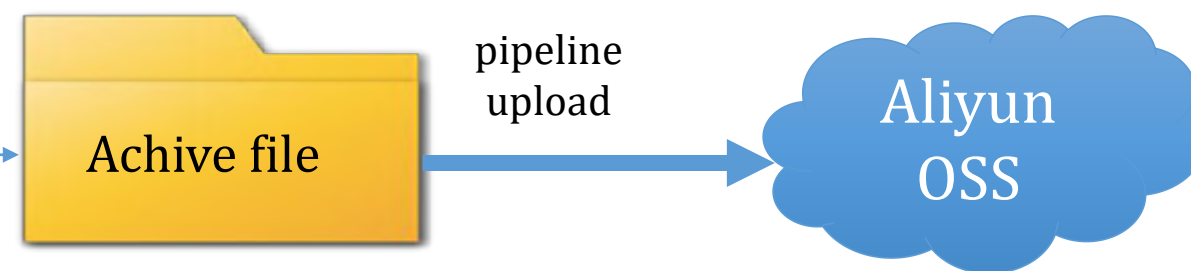


物理备份流程



MongoDB
dbpath

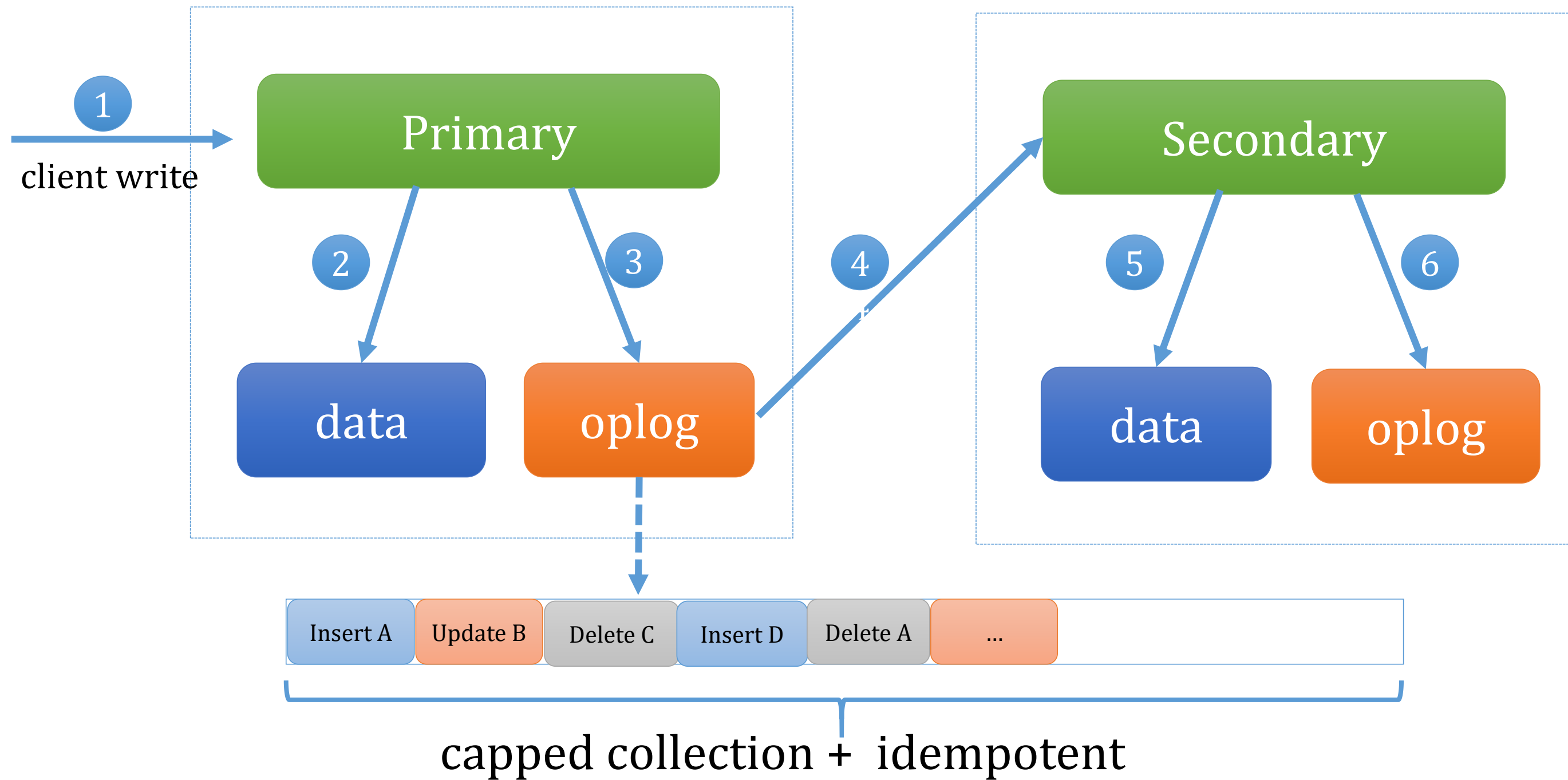
- 拷贝数据目录所有文件，效率高
- 备份、恢复快
- 对业务影响较小
- 跟数据库版本、配置强关联



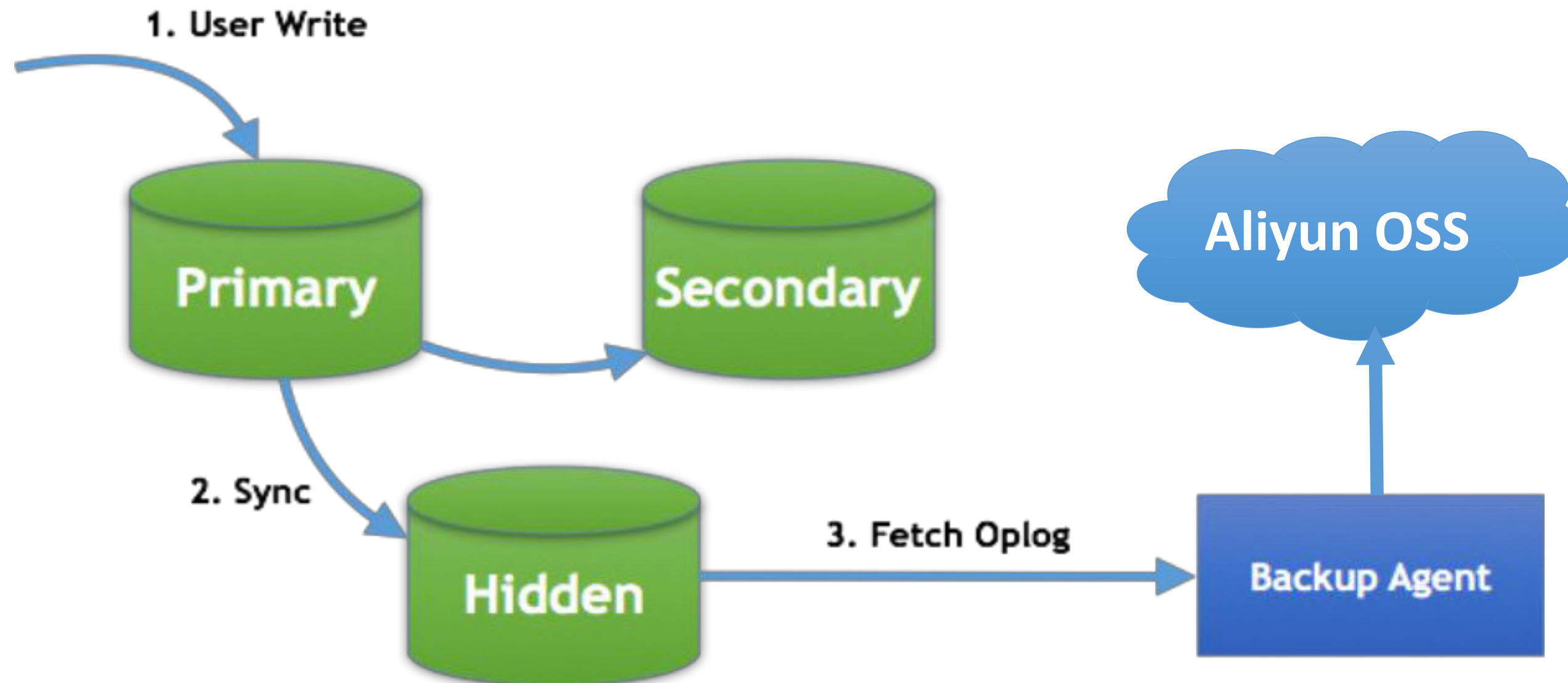
逻辑备份 vs 物理备份

	逻辑备份	物理备份
备份效率	低 数据库接口读取数据	高 拷贝物理文件
恢复效率	低 下载备份集 + 导入数据 + 建立索引	高 下载备份集 + 启动进程
备份影响	大 直接与业务争抢资源	小
备份集大小	比原库小 无需备份索引数据	与原库相同
兼容性	兼容绝大部分版本 可跨存储引擎	依赖存储布局

增量备份原理



增量备份



- backup agent 持续拉取新的 oplog
- oplog tailable cursor

恢复至任意时间点

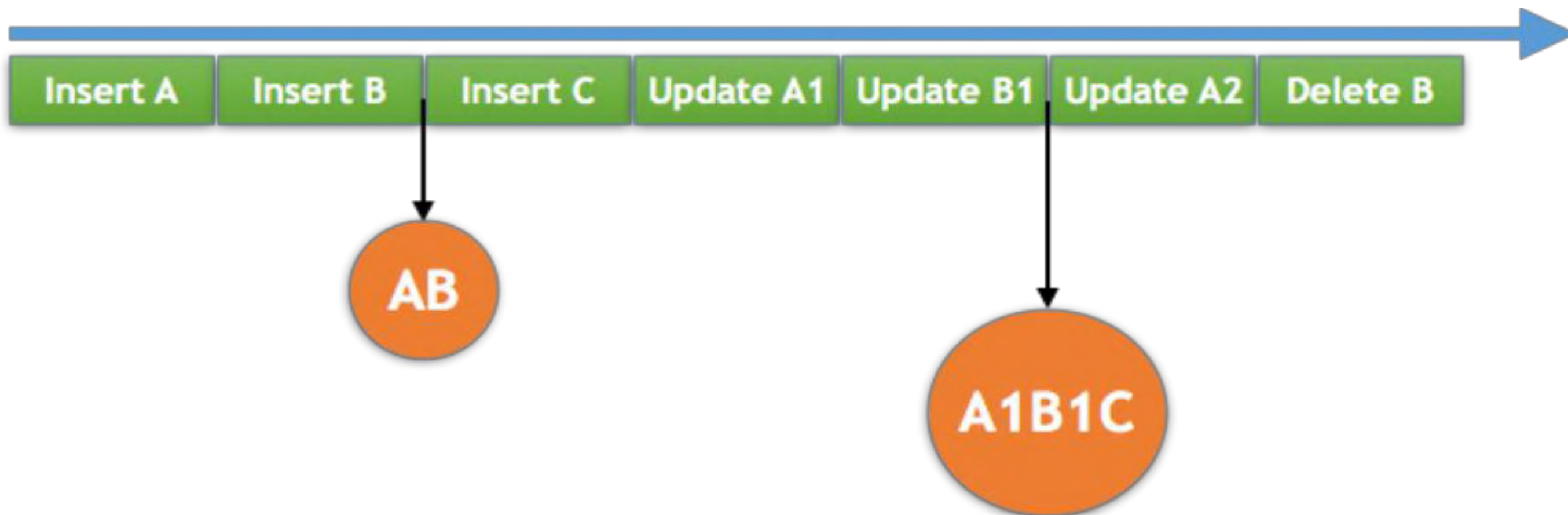
全量备份

+

增量备份

=

任意时间点备份



问题与挑战

可正确恢复数据

+

对应到某个时间点

有效备份集2要素

全量逻辑备份如何对应到时间点？

全量物理备份如何对应到时间点？

增量备份如何确保拉到所有的oplog？

如何确保备份集可恢复？

如何处理备份过程中的异常？

解决方案

逻辑备份

- 支持在线修改oplog，配合 mongodump --oplog
- `db.runCommand({ collMod: "oplog.rs", maxSize: 1024000000 })`

物理备份

- 传统方法：移除或锁定secondary节点停写备份，加回去同步可能追不上
- 改进方法：修改wiredtiger引擎，支持在线热备份，不影响写入

增量备份

- 修改MongoDB源码支持设置 oplogDeleteGuard
- `db.runCommand({ collMod: "oplog.rs", oplogDeleteGuard: 1400000000 })`

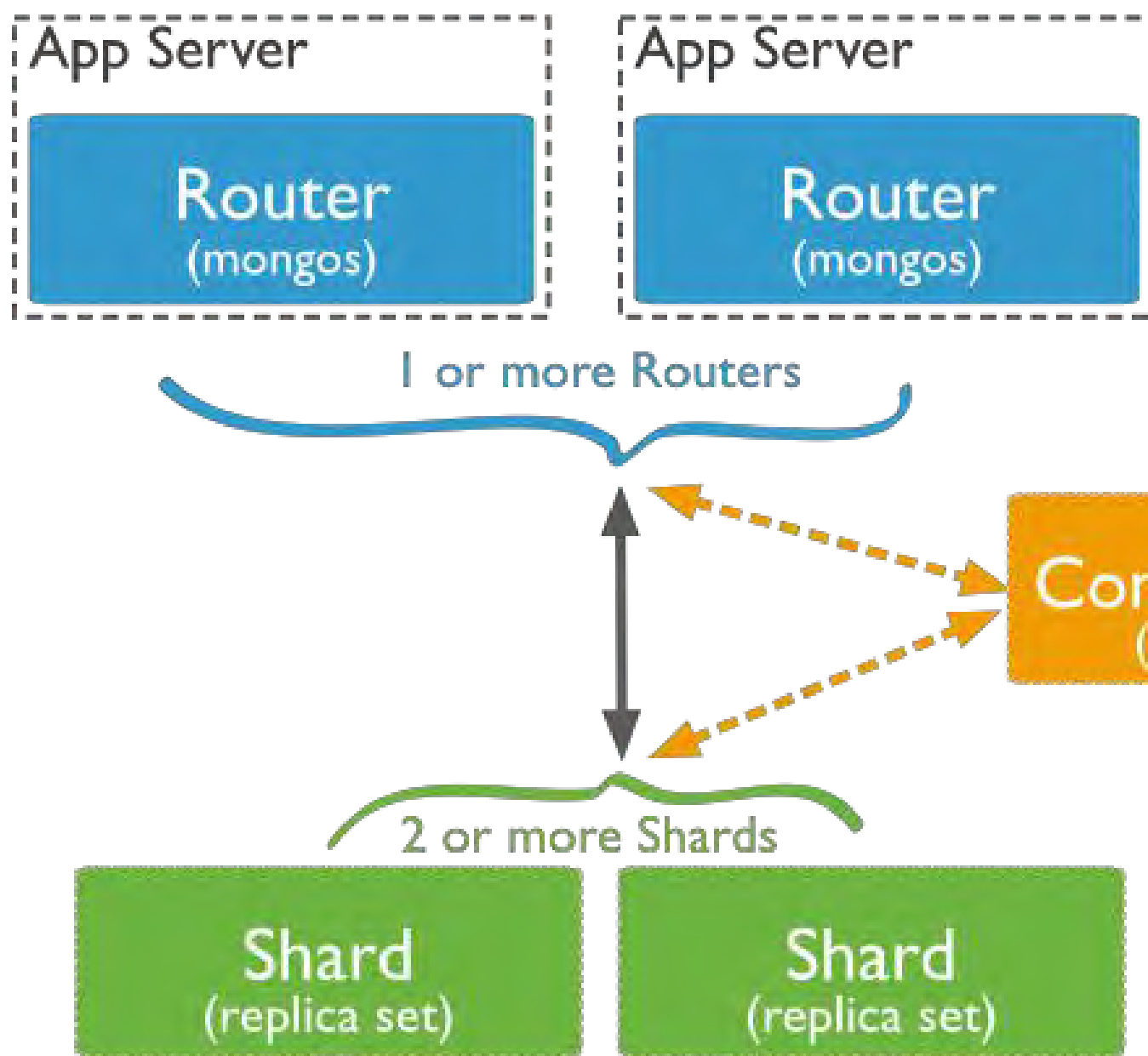
备份验证

- MD5校验避免传输过程中错误
- 定期抽检实例备份集，验证备份集是否可恢复

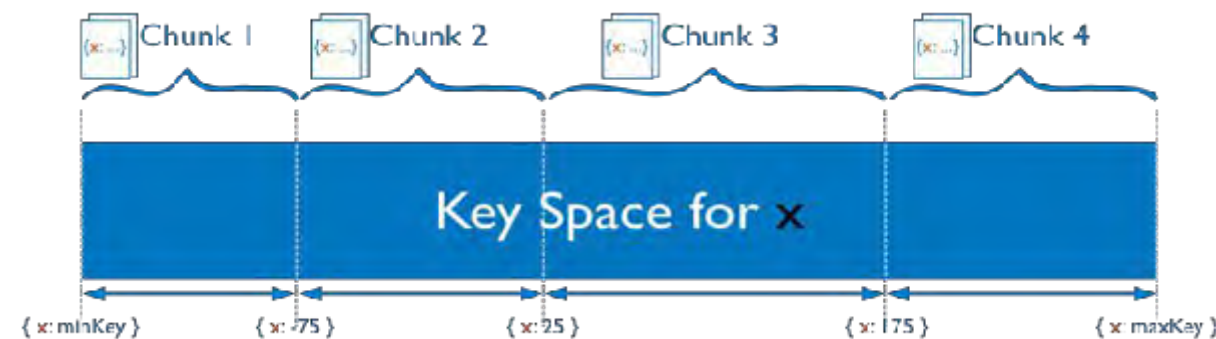
异常处理

- 备份失败时自动failover重试
- hidden节点故障时，到secondary备份，最后尝试primary

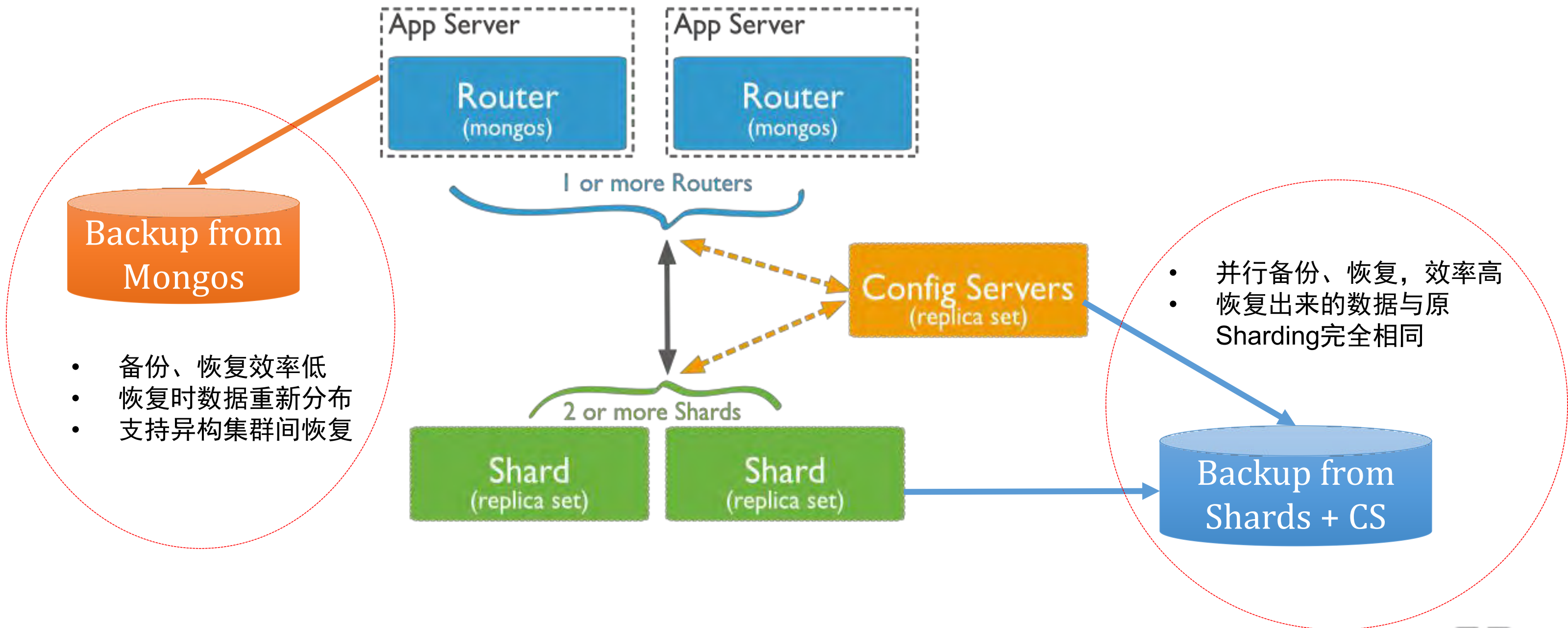
MongoDB Sharding



- mongos负责路由，无状态
- Config Server 存储元数据
- Shard 存储实际用户数据
- 数据根据shardKey分散到各个shard
- 自动在shard间迁移数据做负载均衡



Sharding备份策略



Backup from Mongos

- 备份、恢复效率低
- 恢复时数据重新分布
- 支持异构集群间恢复

- 并行备份、恢复，效率高
- 恢复出来的数据与原 Sharding完全相同

Backup from Shards + CS

Sharding备份挑战

Shard恢复
到任意时间点

+

CS恢复
到任意时间点

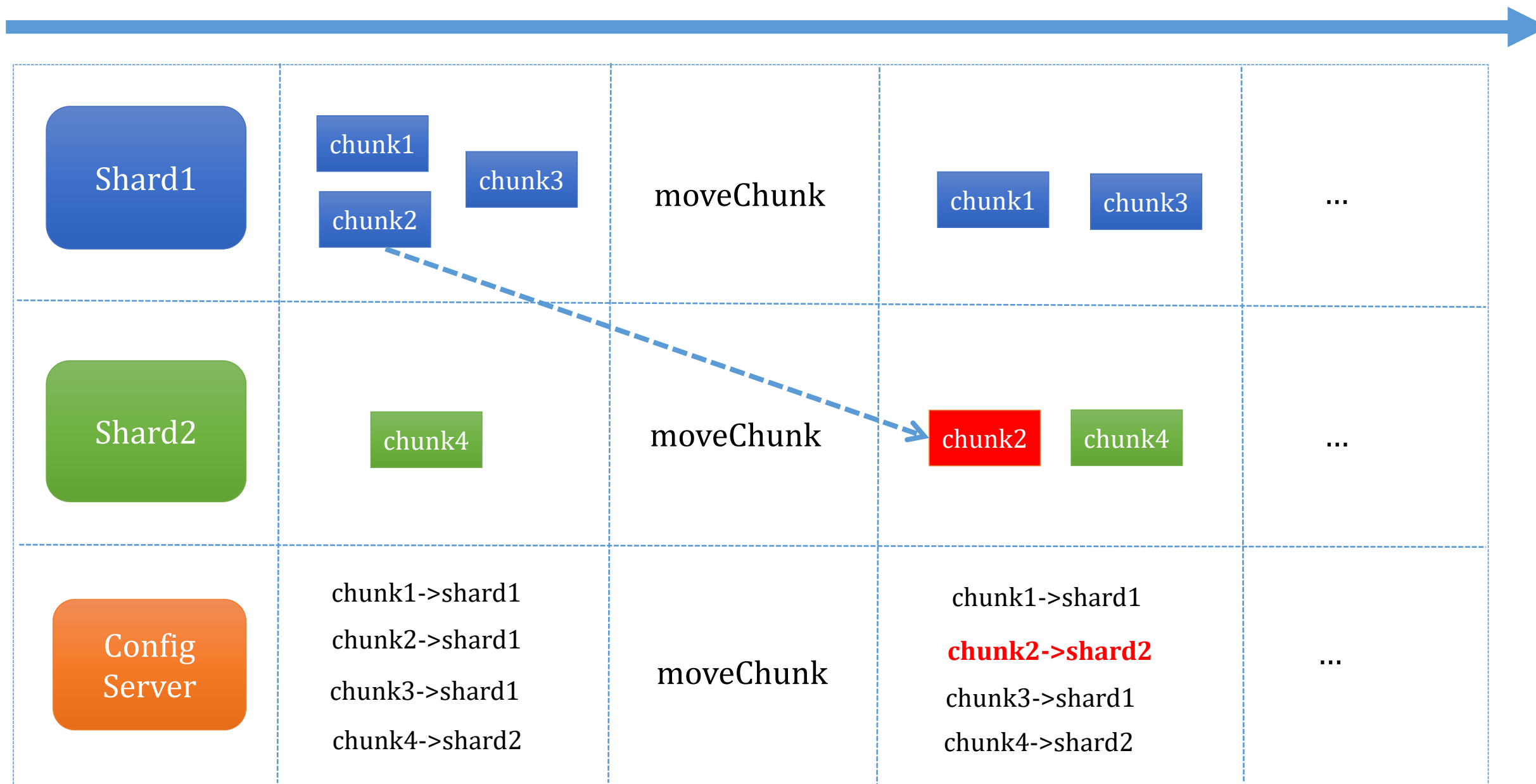
=

?

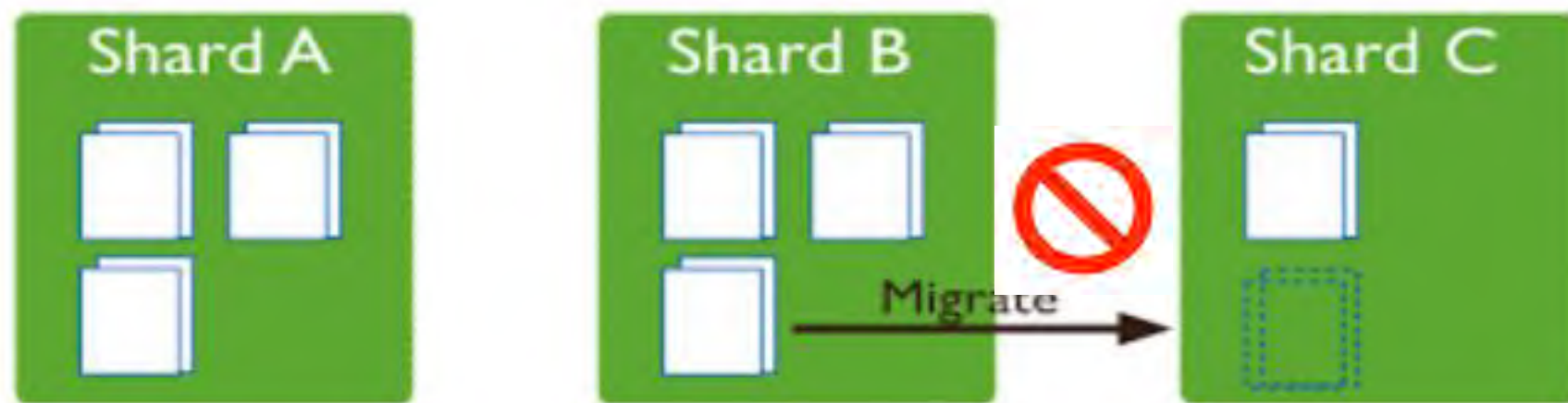
Sharding恢复到
任意时间点备份

Sharding 备份挑战 - 自动负载均衡

时间序

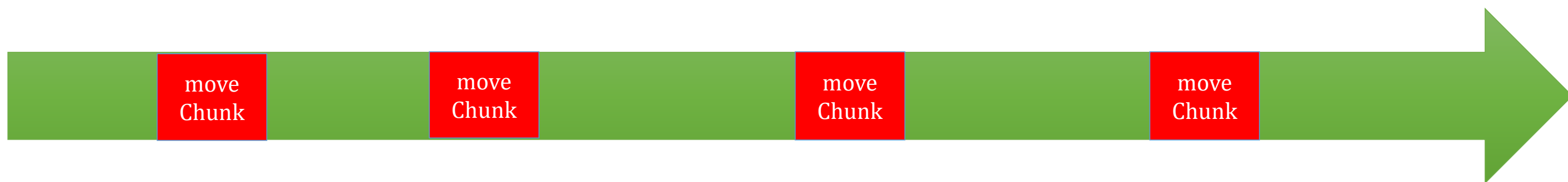


解决方案



传统方案

- 停止 Balancer
- 负载可能不均衡出现热点



改进后方案（节点间时钟误差不能太大）

- 开启 Balancer
- 分析 Config server 迁移日志
- 恢复时避开 chunk 迁移的时间区间

备份产品形态

- 支持恢复到任意时间点
- 支持实例覆盖性恢复
- 支持根据备份集创建新实例
- 备份集可下载，方便迁移

备份与恢复

备份列表 备份设置

选择时间范围 至

备份开始/结束时间	备份策略	备份大小	备份方法	备份类型	状态	操作
2017-03-16 16:37:49 / 2017-03-16 16:39:29	手动备份	1.00K	逻辑备份	全量备份	完成备份	<input type="button" value="下载"/> <input type="button" value="从备份点创建实例"/> <input type="button" value="数据恢复"/>

备份与恢复

备份列表 备份设置

配置信息

保留天数: 备份周期:

备份时间:

- 默认保留最近7天备份
- 备份时间段、周期可配置



有备无患

SDCC 2017 | 上海

数据库核心技术与应用实战峰会

CSDN