



Open Ta//k OpenResty x Open Talk
CHENGDU 2019

Openresty 在又拍云存储中的应用

黄励博@huangnauh

August 31, 2019

主办方：  OpenResty  又拍云

分布式存储特点

- **高可用**

多个无状态代理 分布式数据节点

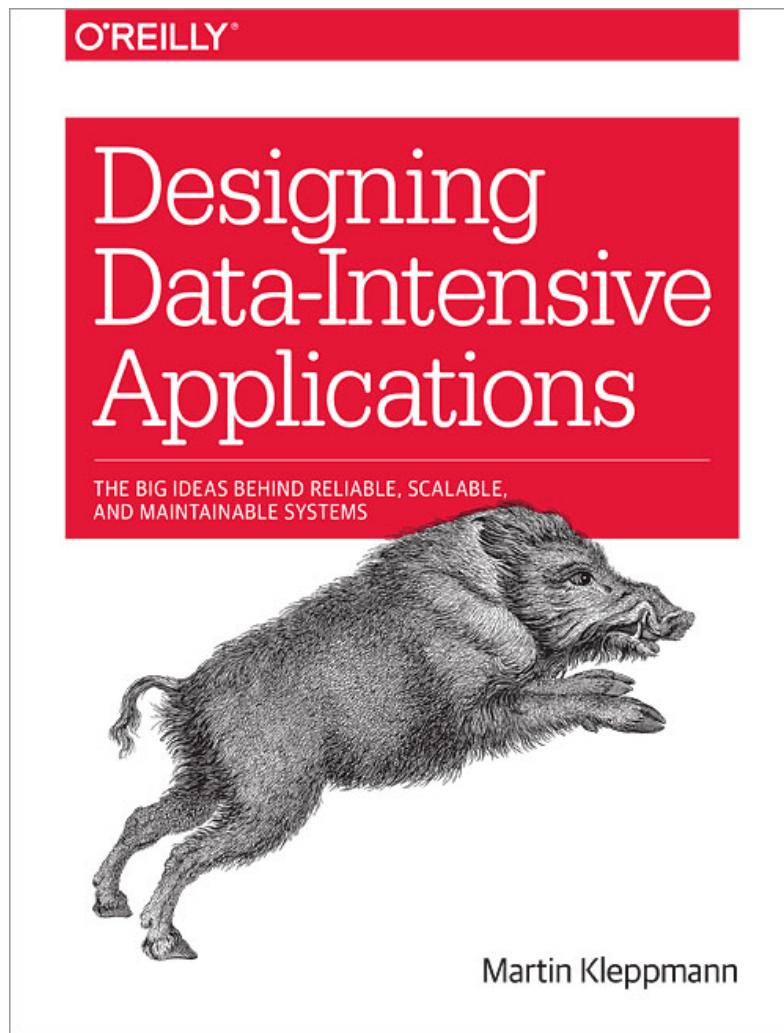
- **易扩展**

有简单的办法应对系统的增长

- **易维护**

在各个服务的不同生命周期，都能方便地进行各项调整

推荐



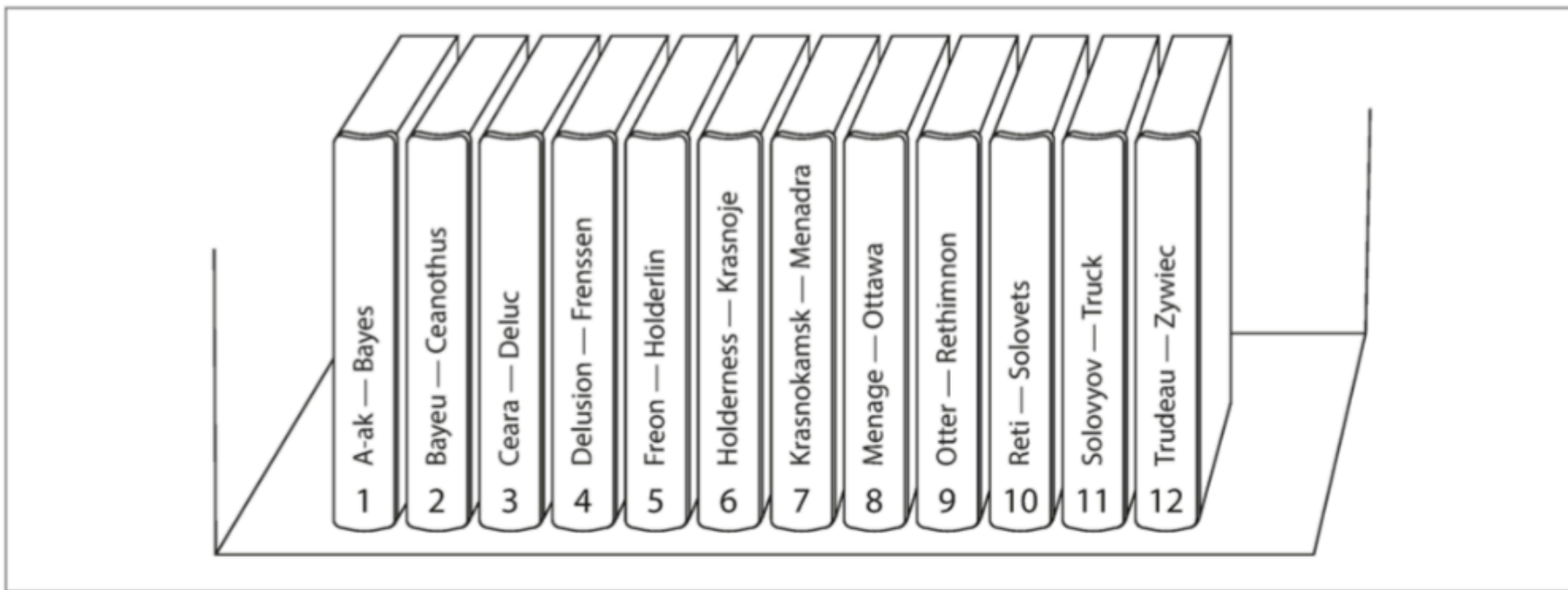
设计数据密集型应用

存储数据I：拆分

分区

将大型的数据拆分成较小的子集

- 按 key 顺序，方便 Range 操作
- 按 key hash，分散访问



又拍云存储系统

由 OpenResty 完成数据拆分



索引和内容拆分

文件路径(key) 与 用户数据(value) 解耦

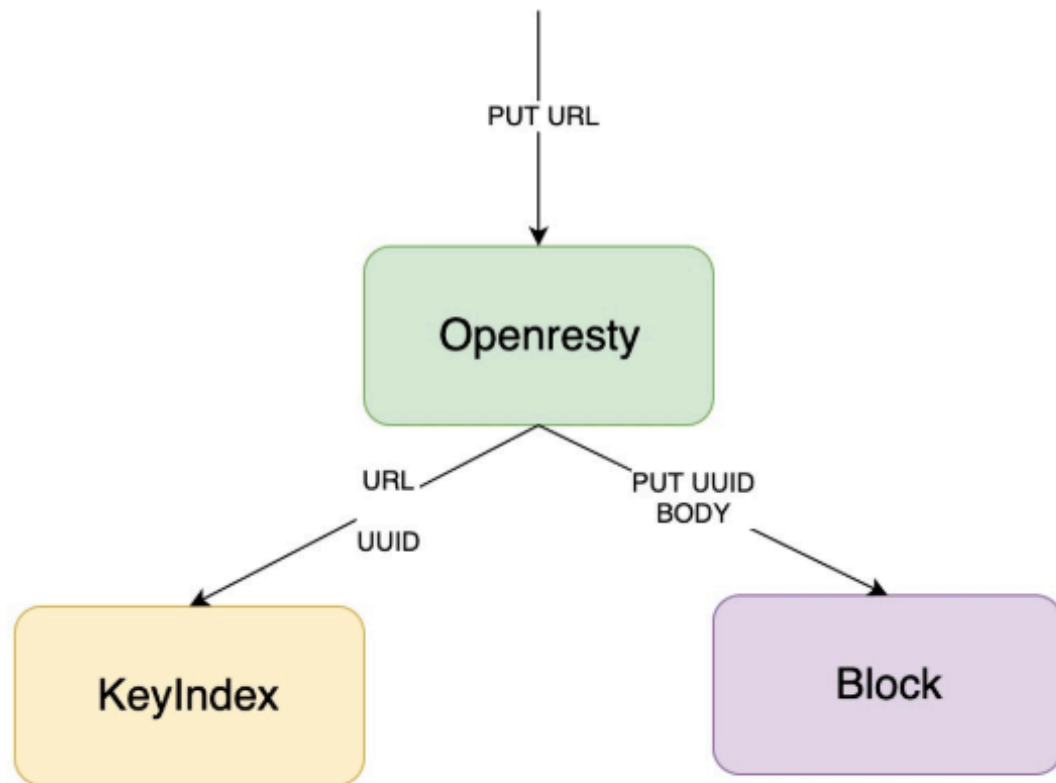
`https://example.com/2019/07/03/video-1562123303.mp4`



`bb17414f-6161-439f-89e9-b3e7dec2a57c`

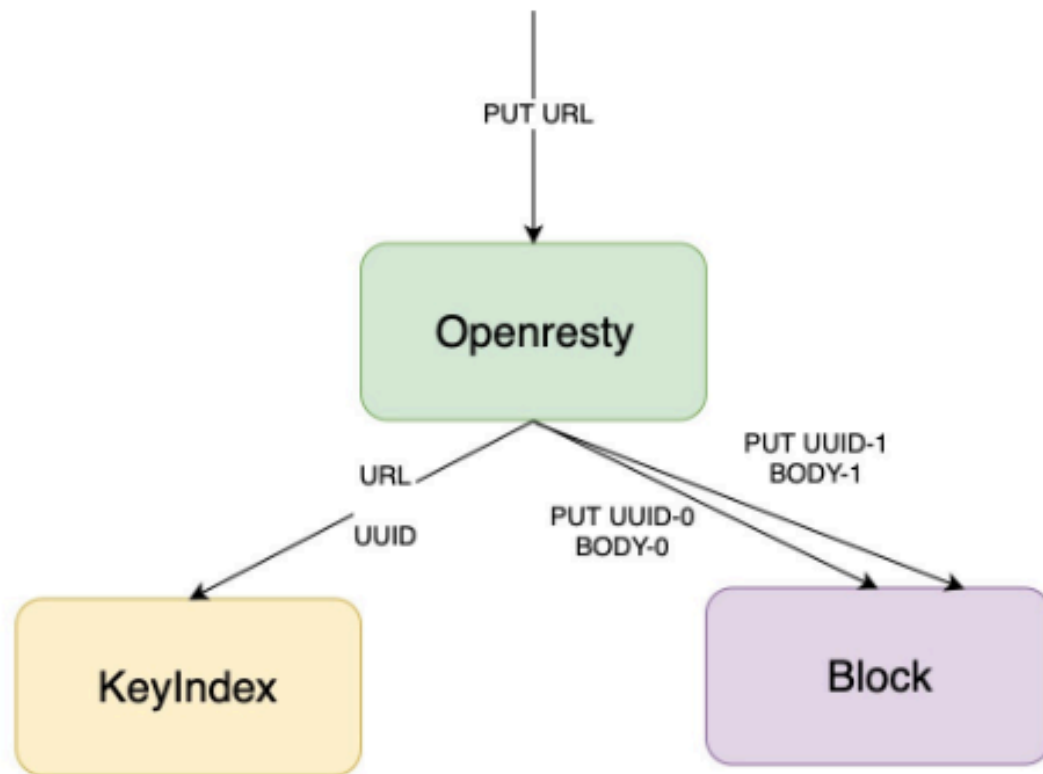
防止热点数据，多数 LSM tree 文件的组织形式是分层有序

索引和内容拆分



Meta 集群依然会有热点，但是由于 value 小，容易 balance
Block 集群的 key 和文件路径无关

内容内部拆分



目的1：一个大文件可以对应多个 Value，分布不同节点和磁盘，负荷分担

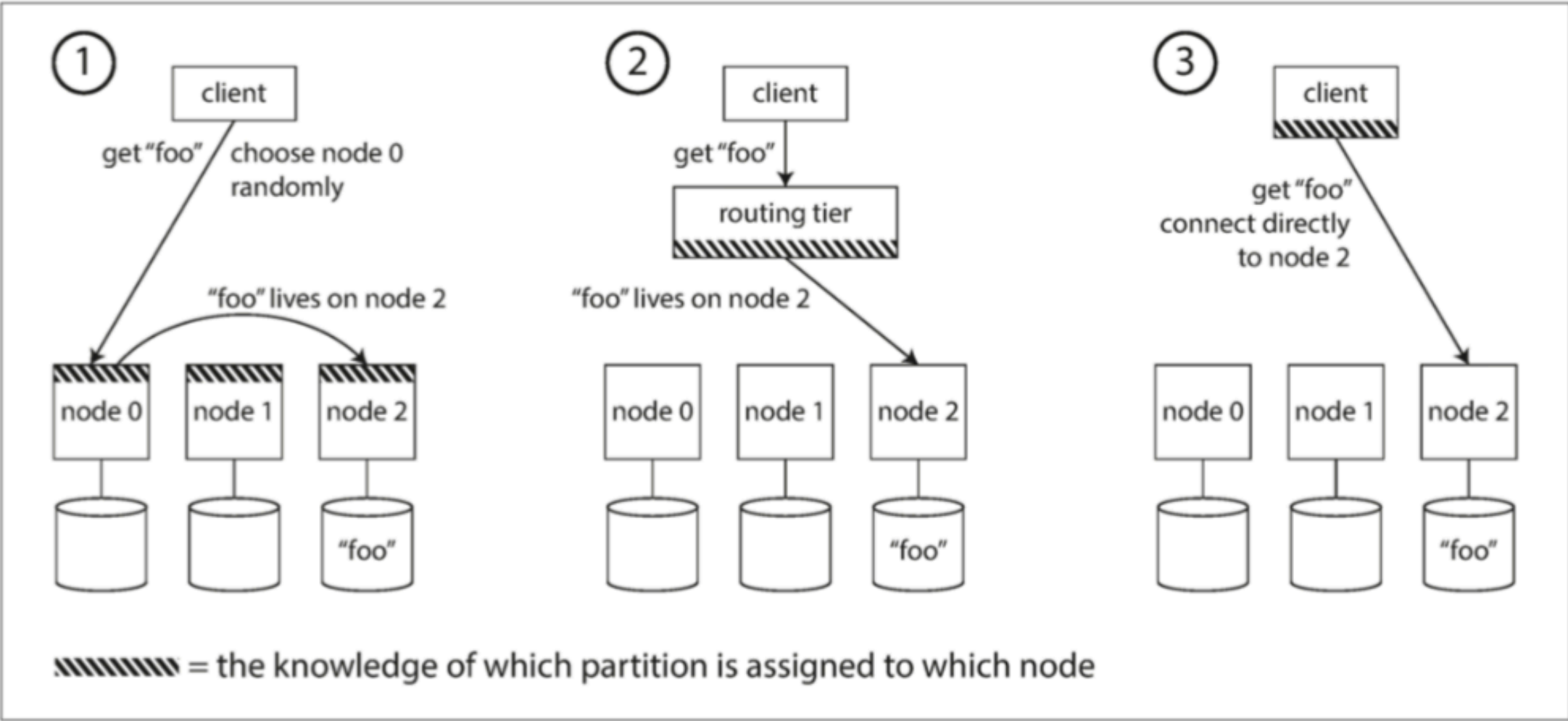
目的2：方便一些不支持 Range 操作的 Block 集群

拆分

- 第一次拆分， 文件路径(uri) 对应多个 Meta 集群，固定分区
- 第二次拆分， 一个 Meta 集群对应多个 Block 集群
- 第三次拆分， Meta 和 Block 子系统内部分区，节点+磁盘

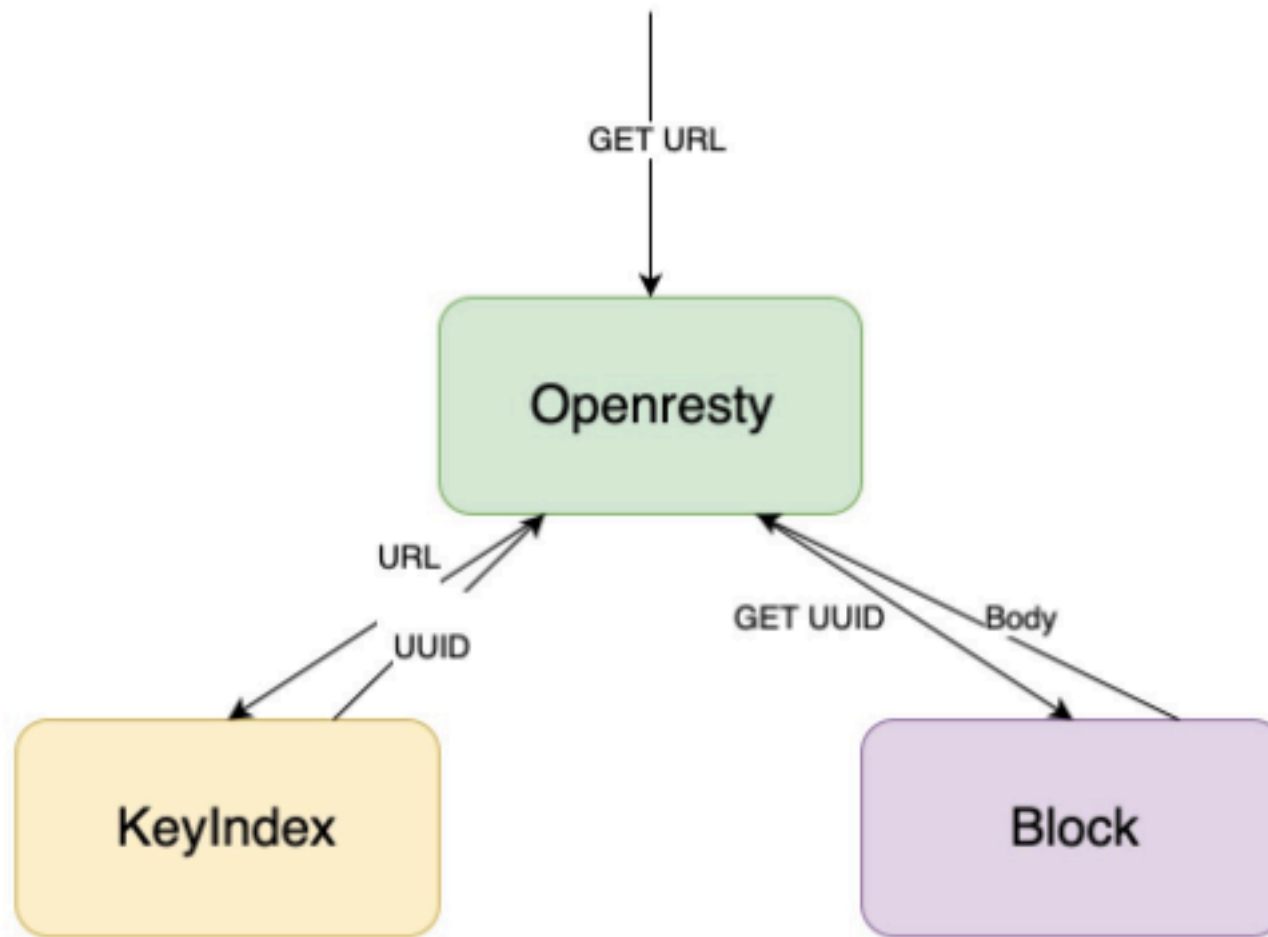
存储数据II：路由

存储数据II：路由



DDIA 第六章截图

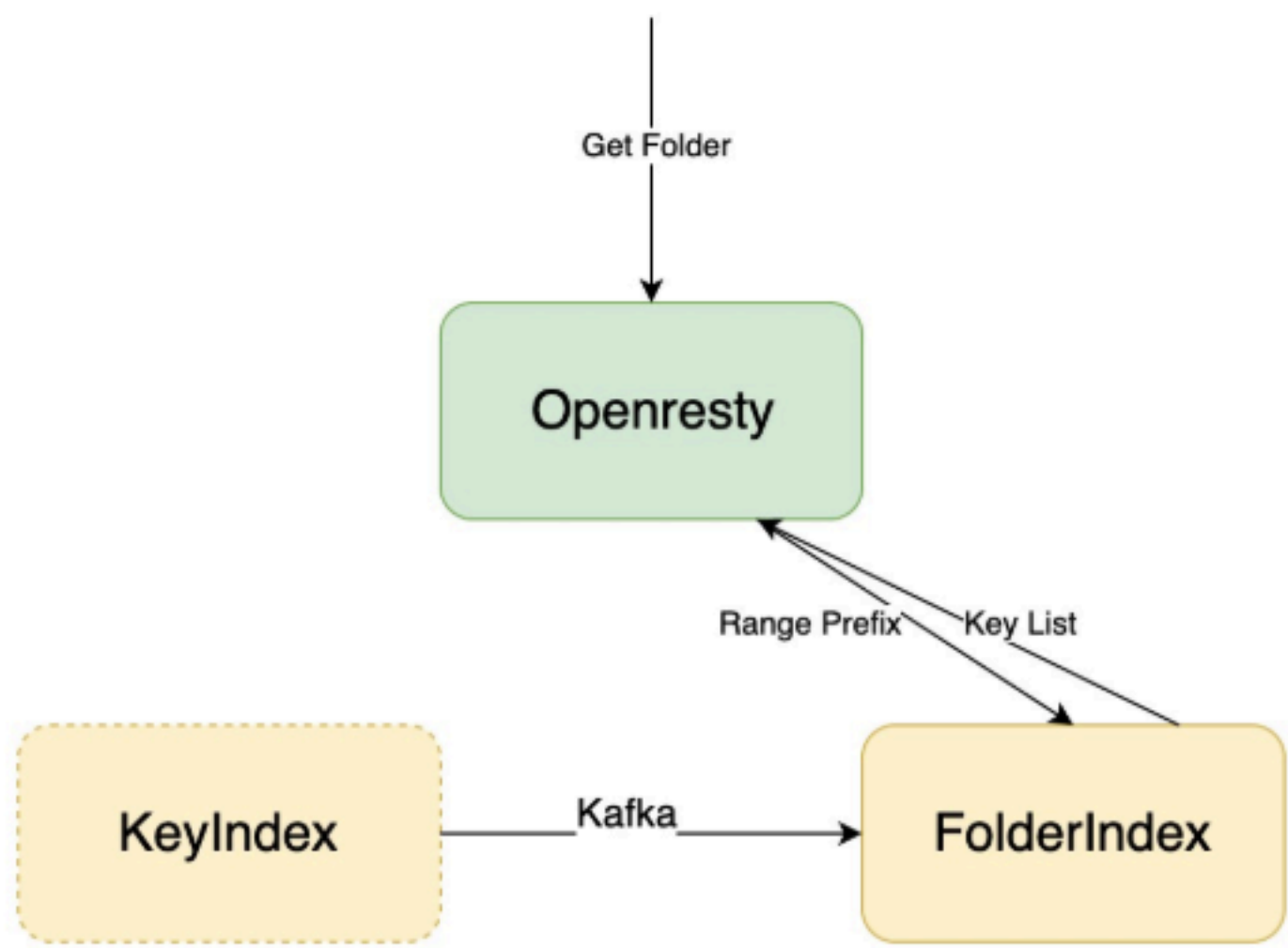
OpenResty 作为路由层



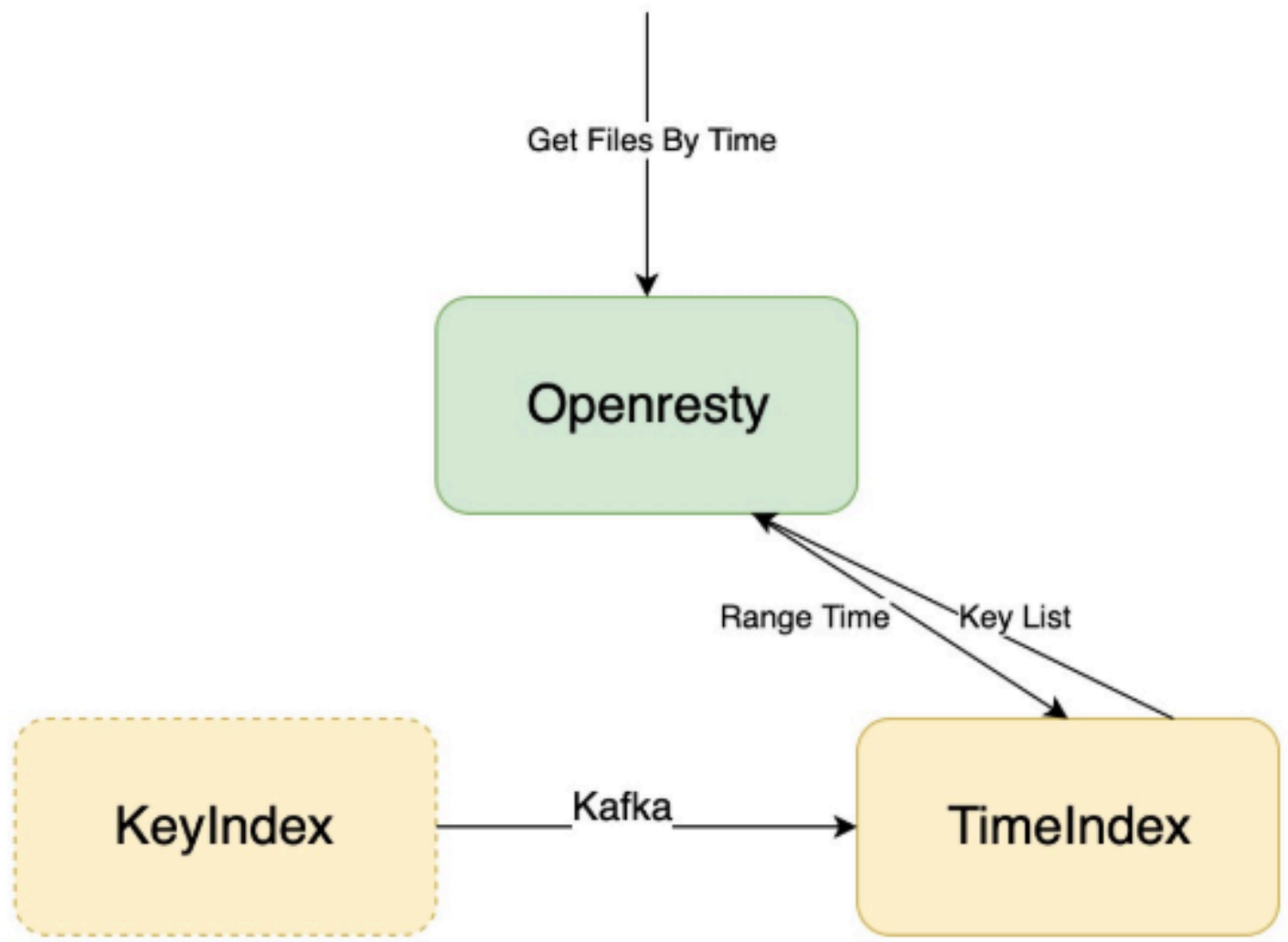
Meta 集群路由: 固定路由

- 不同的用户或者空间
- 不同存储类型
- 不同的索引功能

Meta 路由应用: 目录



Meta 路由应用: 文件按时间过滤

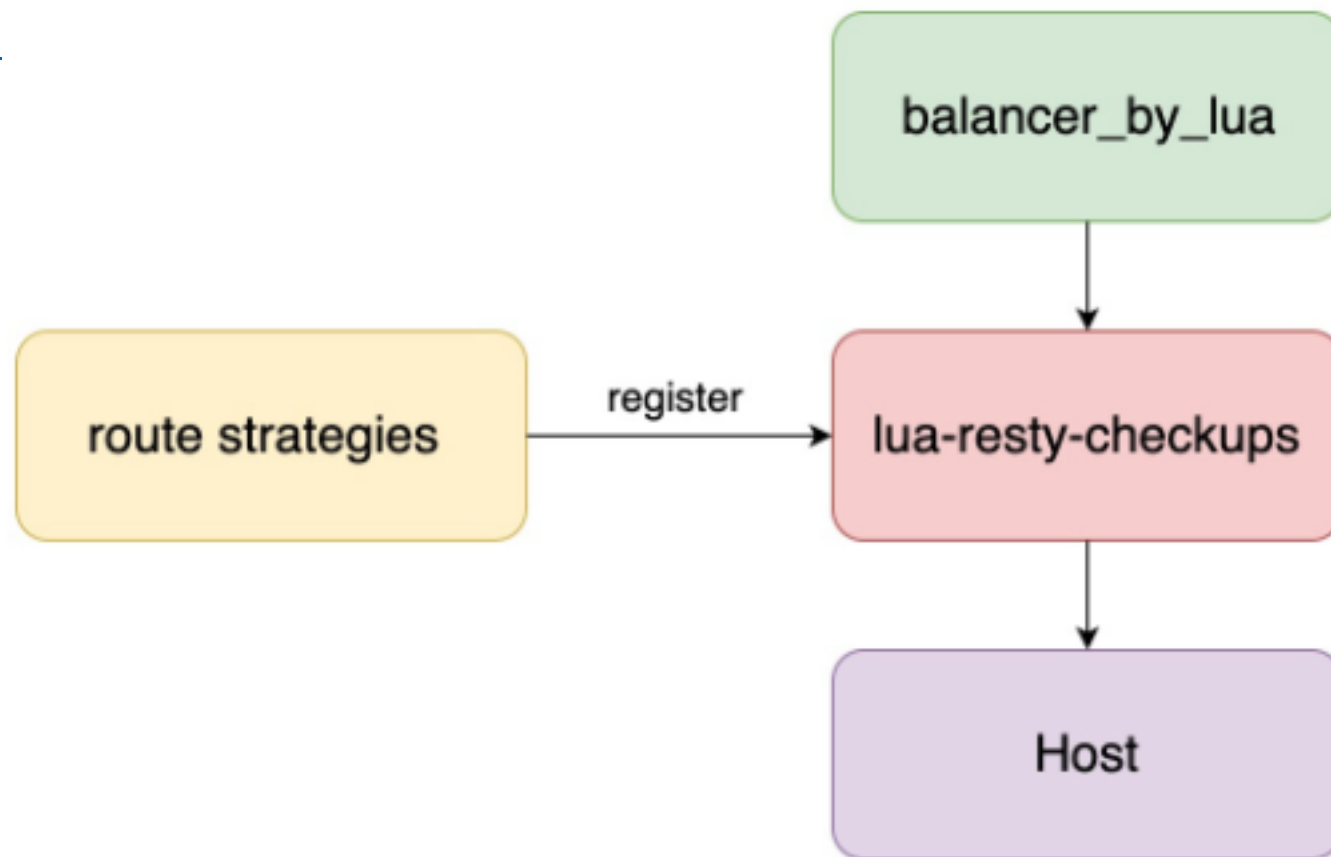


Block 集群路由: 固定或动态路由

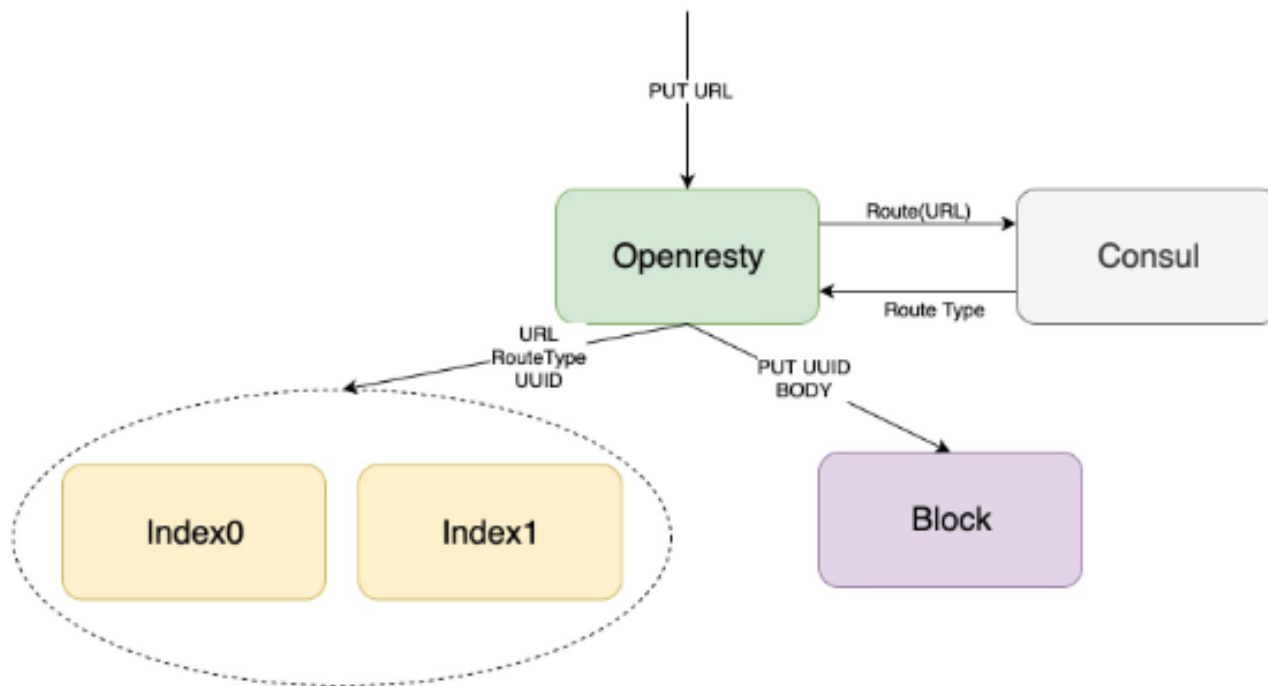
- 和 Meta 集群相同，按用户空间/存储类型划分，固定分区
- 负载均衡，多个同一类型的 Block 集群，按 weight 划分
- 迁移/备份到第三方服务

路由

lua-resty-checkups



路由规则配置



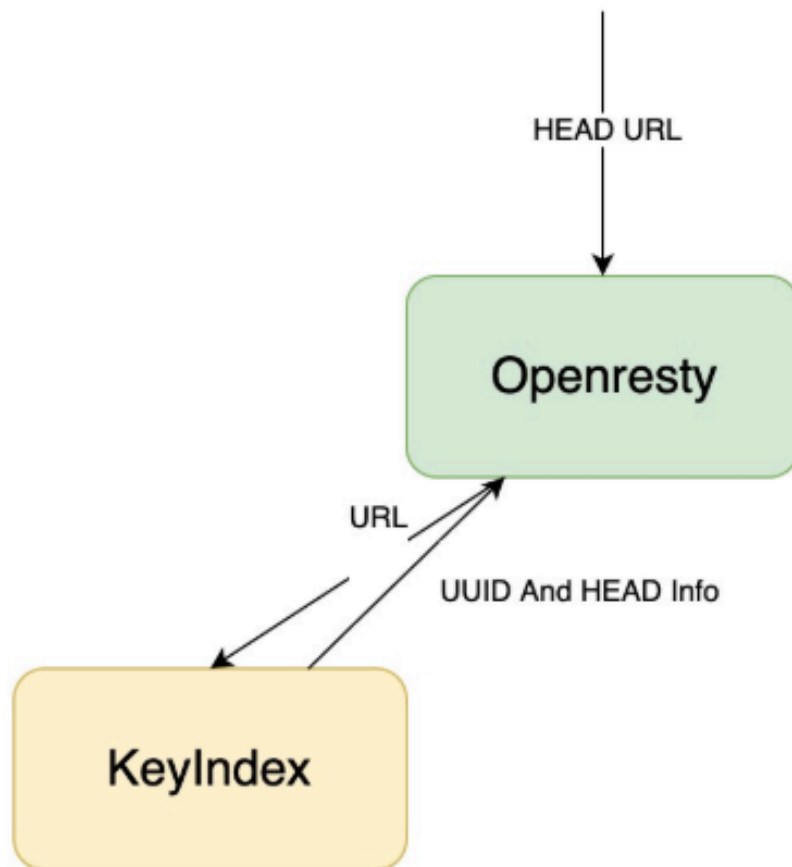
- 定时更新 + 缓存
- 联动更新

配置相关功能采用了 [Slardar](#) 的相关模块

存储数据III: 常用功能

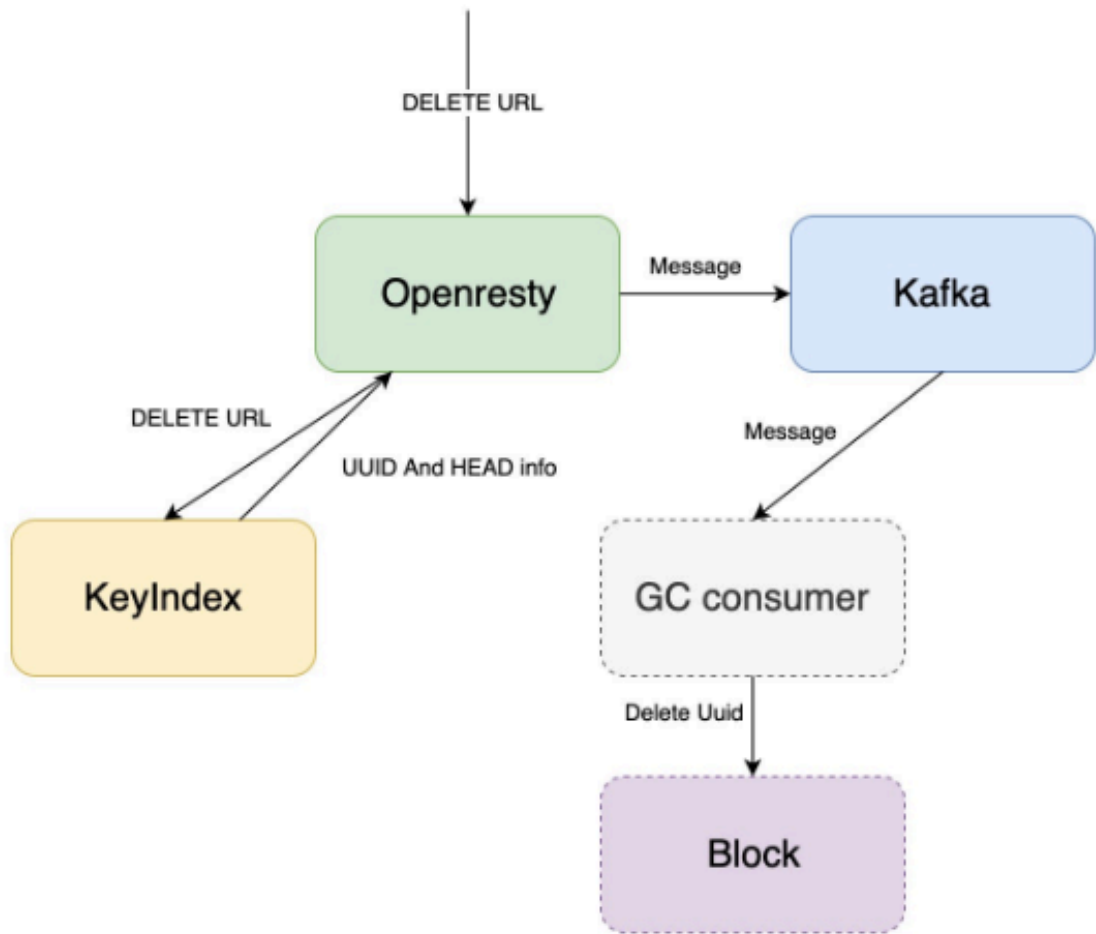
存储数据III: 常用功能

HEAD 只需要 Openresty 路由到 Meta 系统



存储数据III: 常用功能

DELETE 删除 Meta , GC 负责异步删除 Block



存储数据III: 常用功能



存储数据IV: 扩容

存储数据IV: 扩容

- Meta 集群, 扩容 balance 速度快
- Block 集群, 扩容尽量避免大量文件 balance

扩容策略

- 新增 Meta 节点，集群内 balance
- 拆分 Meta 集群，两个集群间同步
- 新增 Block 集群，提高 weight，直到旧集群变为只读集群，避免 Block 集群 balance

扩容之后的操作和集群的管理都是通过 [lua-resty-checkups](#) 来完成

其他V

存储数据: 其他

复制：由各个子系统独立完成

- Meta 集群：Hbase, Postgresql/Mysql
- Meta 备份：数据导出到 kafka
- Block 集群：EC 存储，多副本存储，单副本存储等

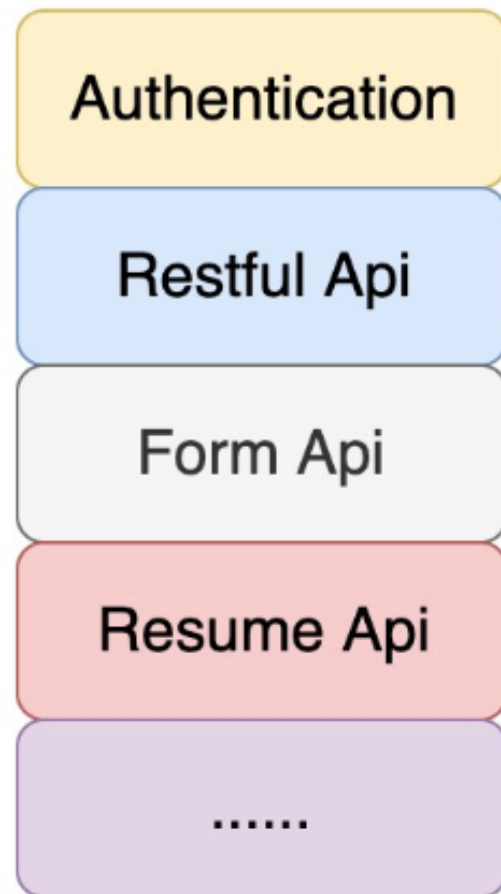
存储数据: 其他

事务：

- 单对象写入
 - 以最后写入 Meta 集群为准 (Last Write Win)
 - 没有写入 Meta 的数据，都会被 GC 回收
 - Meta 集群支持单对象 CAS(compare-and-set) 操作
- 不支持多对象写入
- 限制同一 uri 的重复上传和并发上传

其中 Openresty 的限制模块 可以参考 [openresty/limit-traffic](https://openresty.org/en/limit-traffic.html)，又拍云在此基础上增加了 [token bucket](#) 的方法

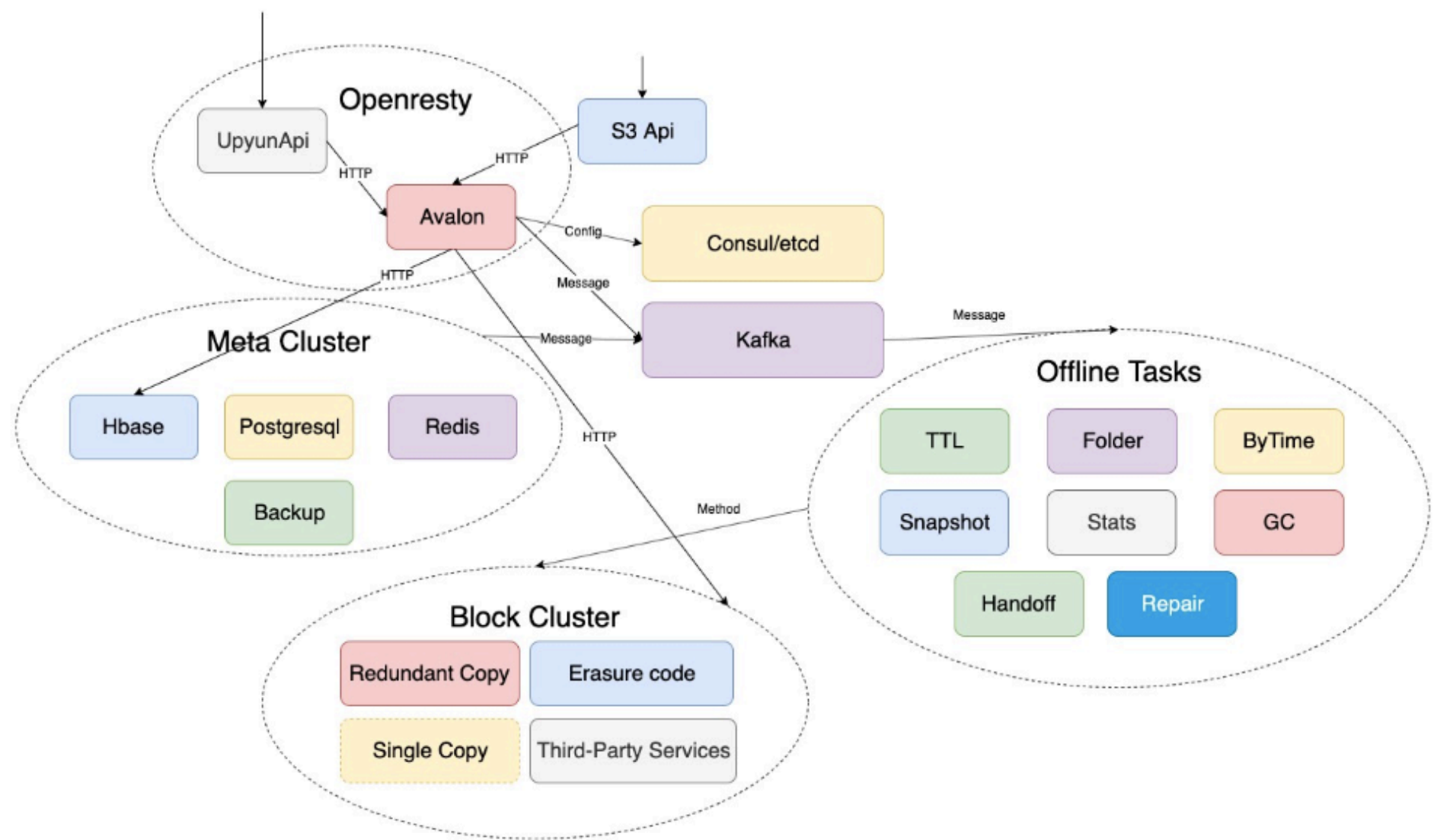
分布式存储之外



断点续传

- 不是所有 Block 系统都支持续传功能
- 续传的每一块文件在 Block 系统都是一个新文件
- 访问的时候，在 OpenResty 中把它拼接起来

又拍云存储系统



OpenResty 相关

[1] [upyun/slardar](#)

[2] [upyun/lua-resty-checkups](#)

[3] [upyun/lua-resty-limit-rate](#)



关注又拍云微信公众号，
获取更多干货！

THANKS !



主办方：  OpenResty  又拍云