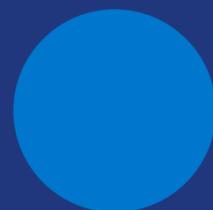




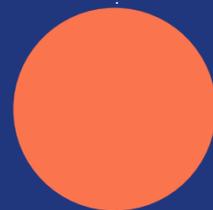
JuiceFS 在 Elasticsearch 的冷热数据分层实践

苏锐 Juicedata 合伙人

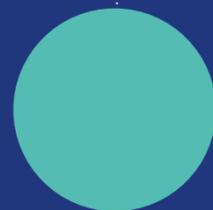
目录



Elasticsearch 的数据分层架构设计



对象存储上使用 Elasticsearch 存在的挑战



JuiceFS 的架构设计及原理解析



JuiceFS 在 Elasticsearch 的实践及案例

Elasticsearch 的数据分层架构设计

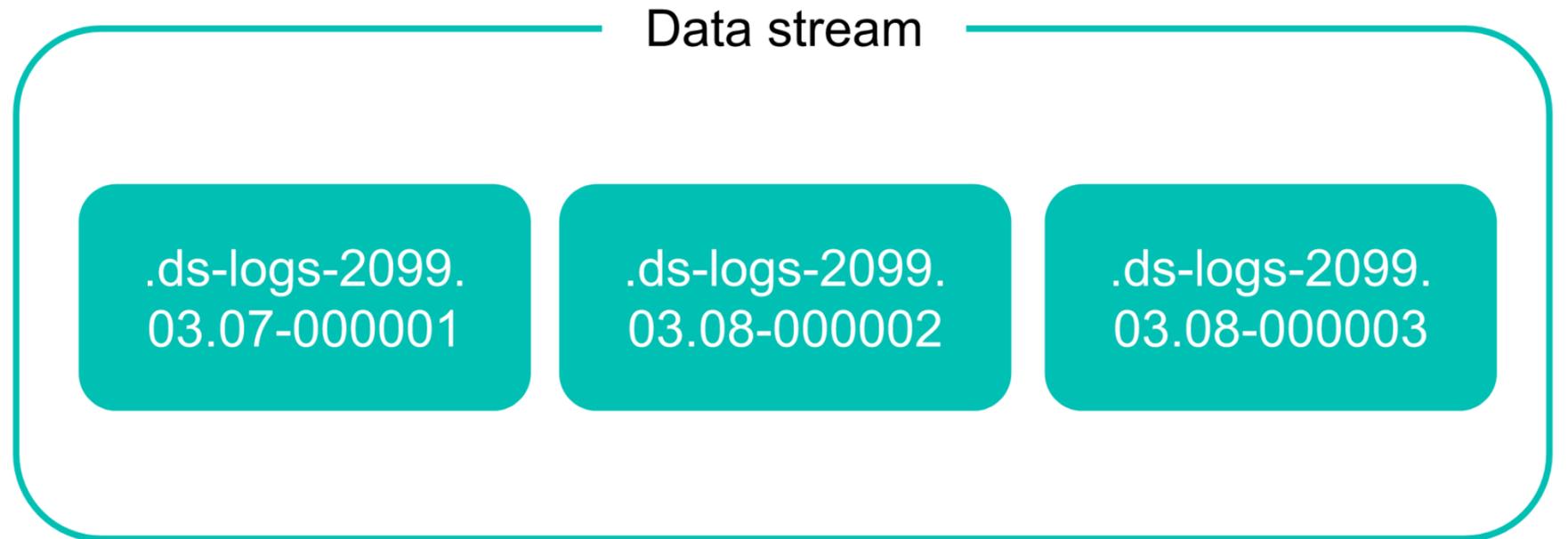


JuiceFS

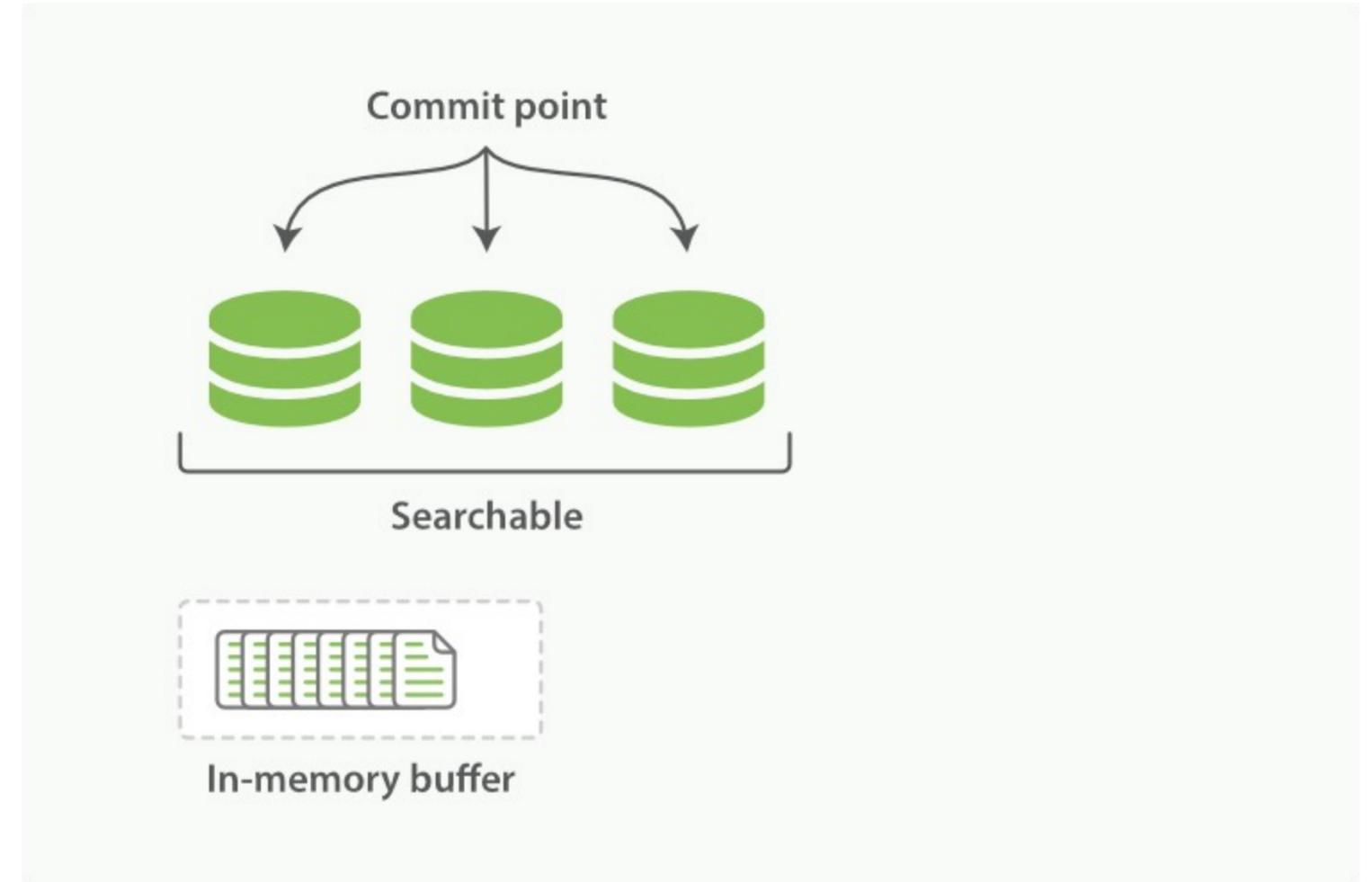
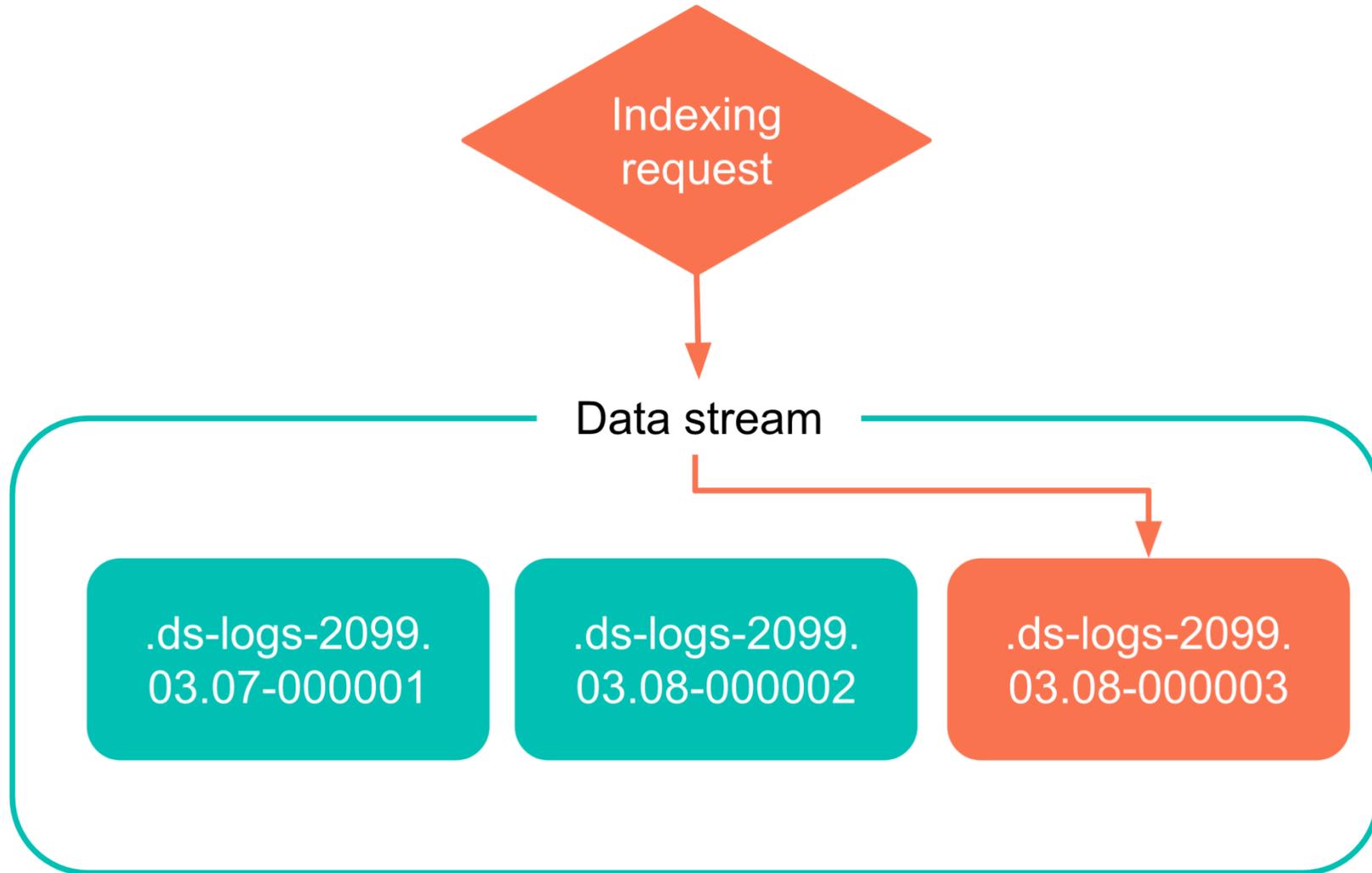


数据流 (Data Stream)

- 流式写入
- 仅追加写
- 必须带有时间戳
- 由多个索引构成
- 典型数据：日志



数据流 (Data Stream)



» Index Lifecycle Management (ILM)

- ILM 定义了索引生命周期的 5 个阶段
- 热数据 (Hot) : 频繁更新和查询的数据
- 温数据 (Warm) : 不再更新, 但仍会被较频繁查询的数据
- 冷数据 (Cold) : 不再更新, 且查询频率较低的数据
- 极冷数据 (Frozen) : 不再更新, 且几乎不会被查询的数据
- 删除数据 (Delete) : 不再需要用到, 可以放心删除的数据

节点角色 (Node Role)

- 为不同 ES 节点分配不同的角色
- 角色基于 ILM 的索引生命周期的不同阶段
- 同一个节点可以有多种角色
- 为不同角色的节点配置不同的存储 (如 SSD、HDD、JuiceFS)
- `node.roles: ["data_hot", "data_content"]`

» 生命周期策略 (Lifecycle Policy)

根据索引的**不同维度特征**（如大小、文档数、**时间**）自动地将索引从某个生命周期阶段滚动（rollover）到另一个阶段

Create policy

[Documentation](#)

Policy name

my-logs

A policy name cannot start with an underscore and cannot contain a comma or a space.

Policy summary

This policy moves data through the following phases. [Learn about timing](#)

Hot phase

Warm phase



Hot phase **Required**

Store your most-recent, most frequently-searched data in the hot tier, which provides the best indexing and search performance at the highest cost.

[Advanced settings](#)



Warm phase

Move data into phase when:

Move data to the warm tier, which is optimized for search performance over indexing performance. Data is infrequently added or updated in the warm phase.

[Advanced settings](#)

Keep data in this phase forever



Cold phase

Move data to the cold tier, which is optimized for cost savings over search performance. Data is normally read-only in the cold phase.

Save policy

Cancel

[Show request](#)



生命周期策略 (Lifecycle Policy)



Hot phase Required

Store your most-recent, most frequently-searched data in the hot tier, which provides the best indexing and search performance at the highest cost.

✓ [Advanced settings](#)

Rollover

Start writing to a new index when the current index reaches a certain size, document count, or age. Enables you to optimize performance and manage resource usage when working with time series data.

Note: How long it takes to reach the rollover criteria in the hot phase can vary. [Learn more](#)

Use recommended defaults

Rollover when an index is 30 days old or reaches 50 gigabytes.

Enable rollover

Maximum index size



Maximum documents

Maximum age



对象存储上使用 Elasticsearch 存在的挑战

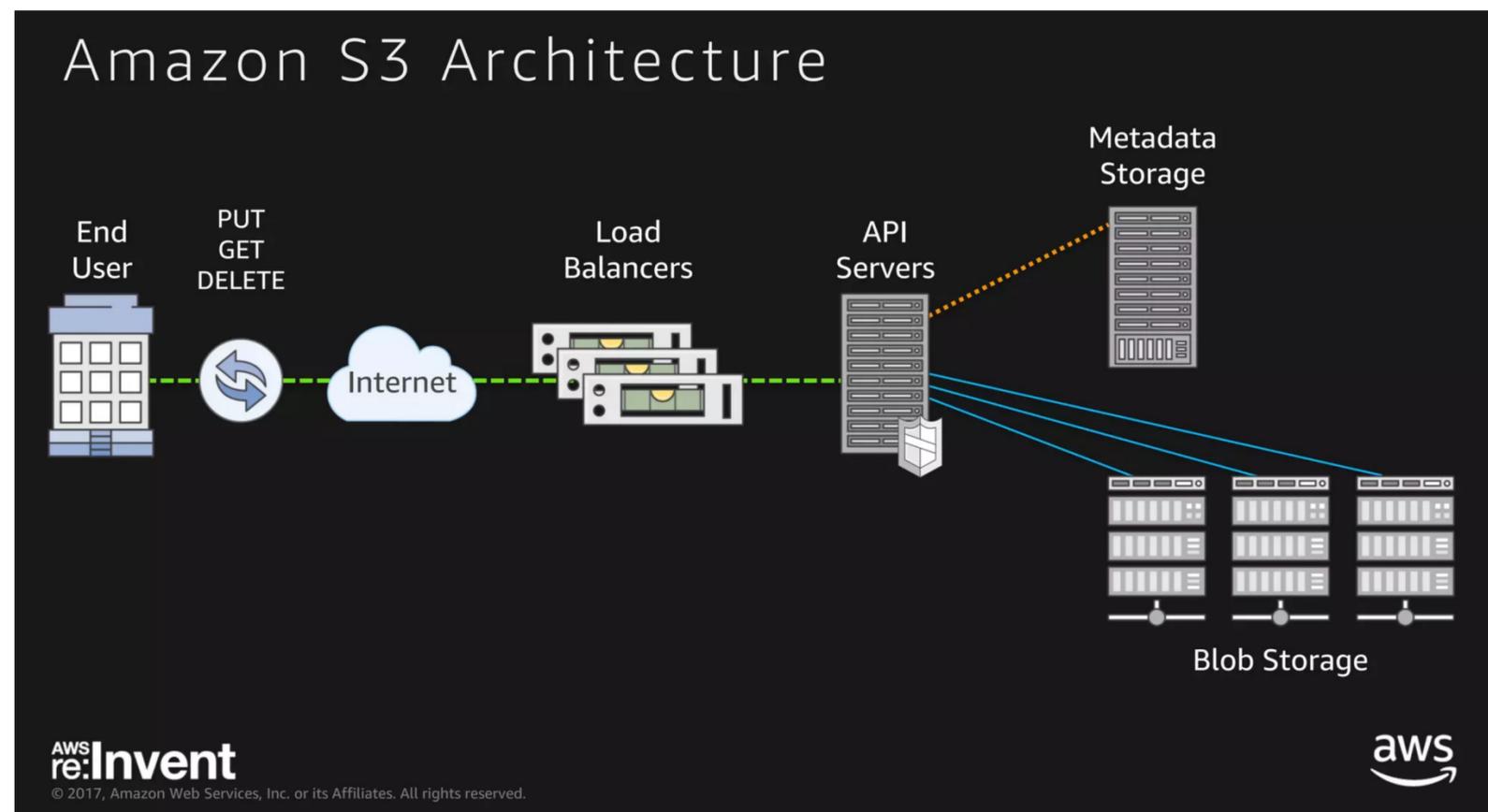


JuiceFS



对象存储架构

- S3 于 2006 年发布
- 以存储海量非结构化数据为目标
- 能支撑万亿级文件数，大小文件均适合
- 低廉的存储成本（支持 EC），可靠的数据持久性（11 个 9）
- 基于 HTTP 协议的 RESTful API
- KV 结构的元数据设计
- 数据不支持修改
- 最终一致性





对象存储 vs. 文件系统

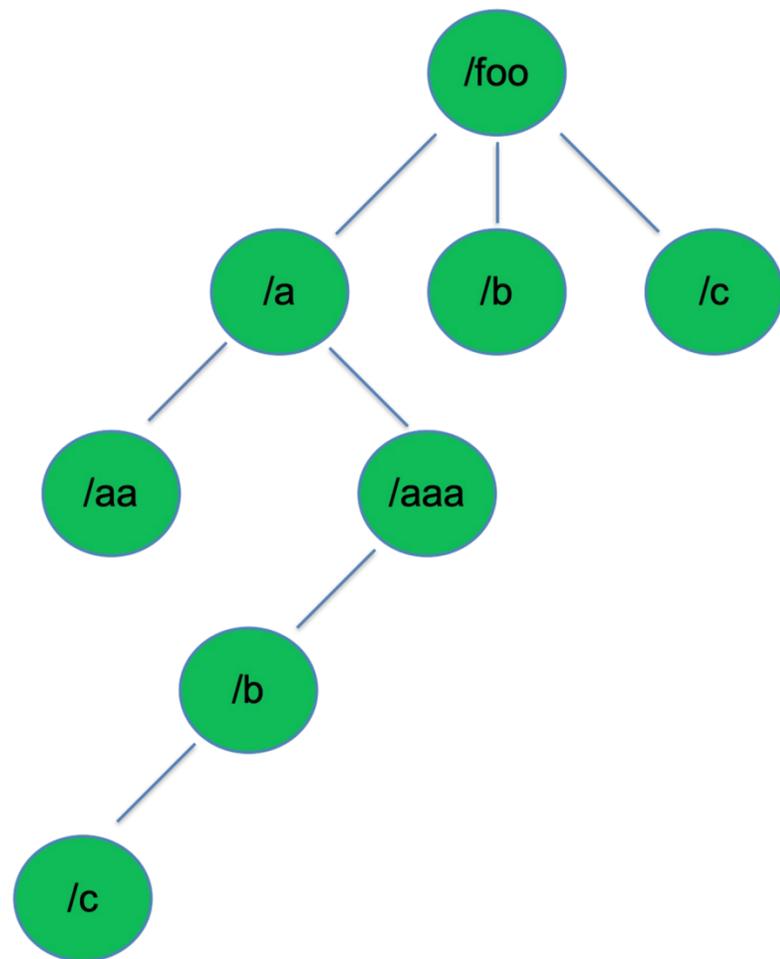
	文件系统	对象存储
存储规模	百亿级（分布式）	★ 万亿级
一致性	★ 强一致性	部分强一致性
容量管理	手动 / 弹性	★ 弹性
原子重命名	★ 支持	不支持
List 性能	★ 高	低
修改数据	★ 支持	不支持
访问接口	POSIX	HTTP

对象存储 vs. 文件系统

如何实现 Rename ?

```
> mv /foo /bar
```

文件系统 📌



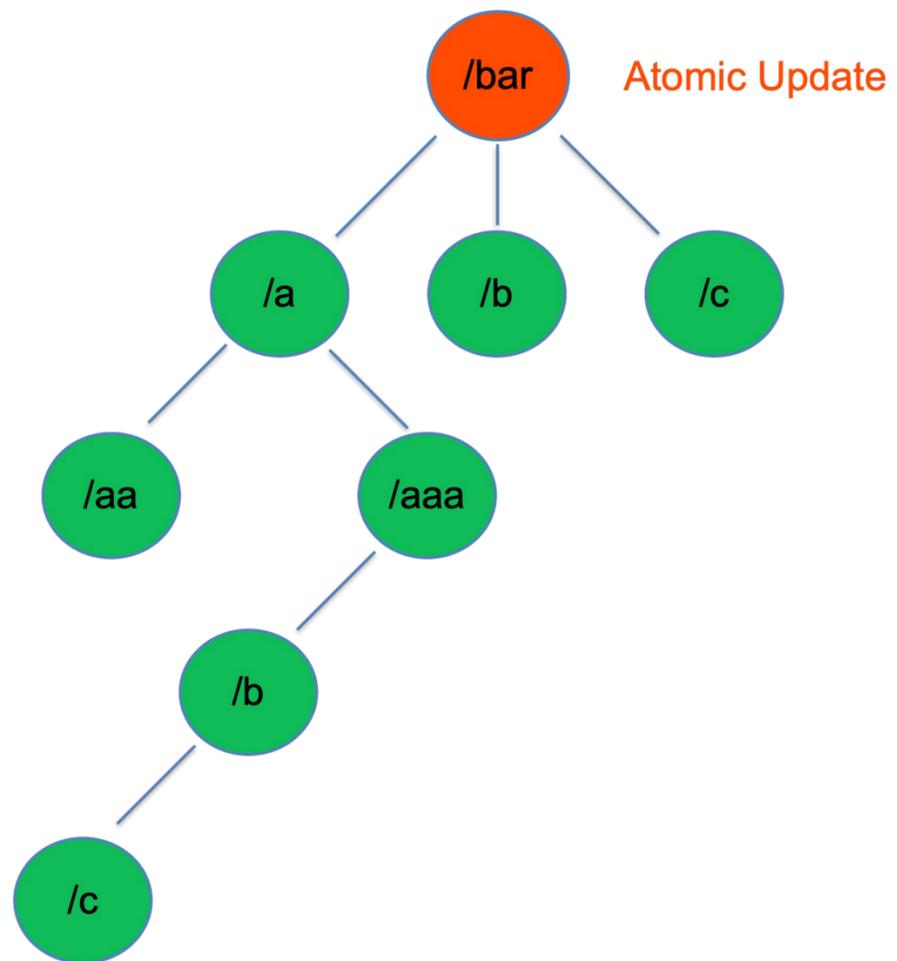
对象存储 📌



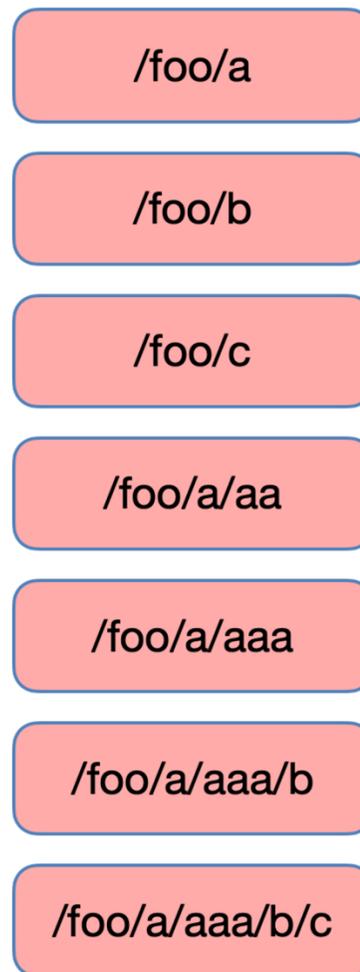
对象存储 vs. 文件系统

如何实现 Rename ? `> mv /foo /bar`

文件系统 📌



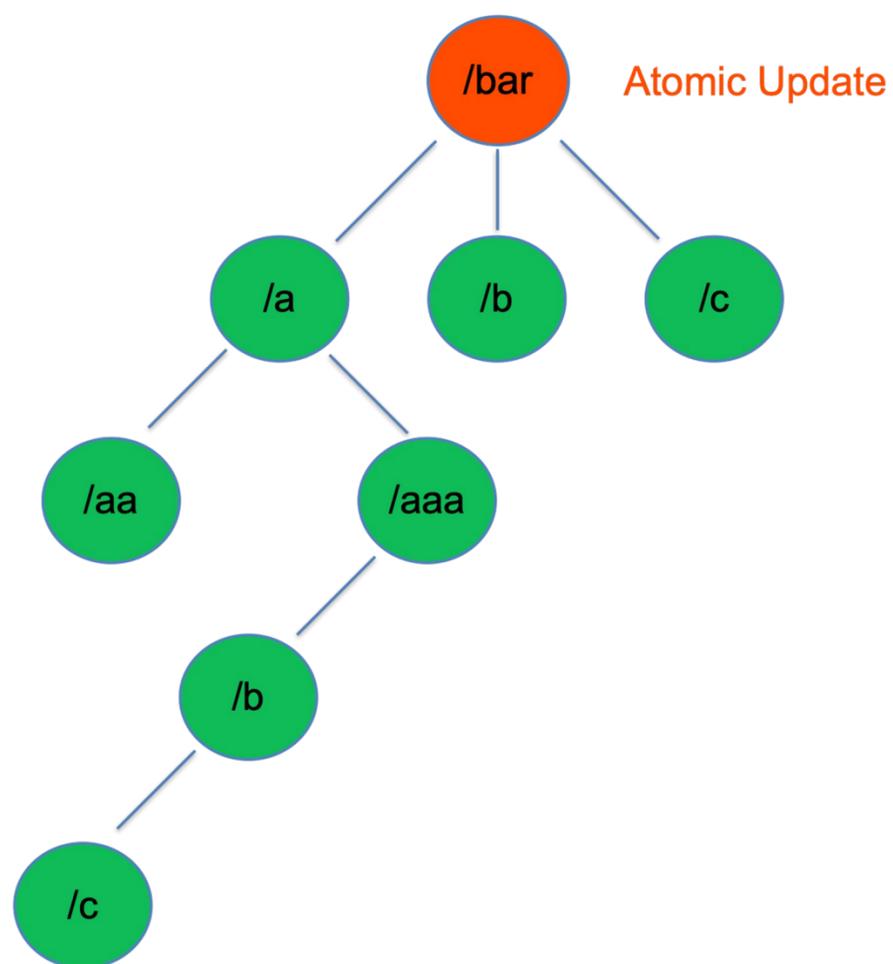
对象存储 📌



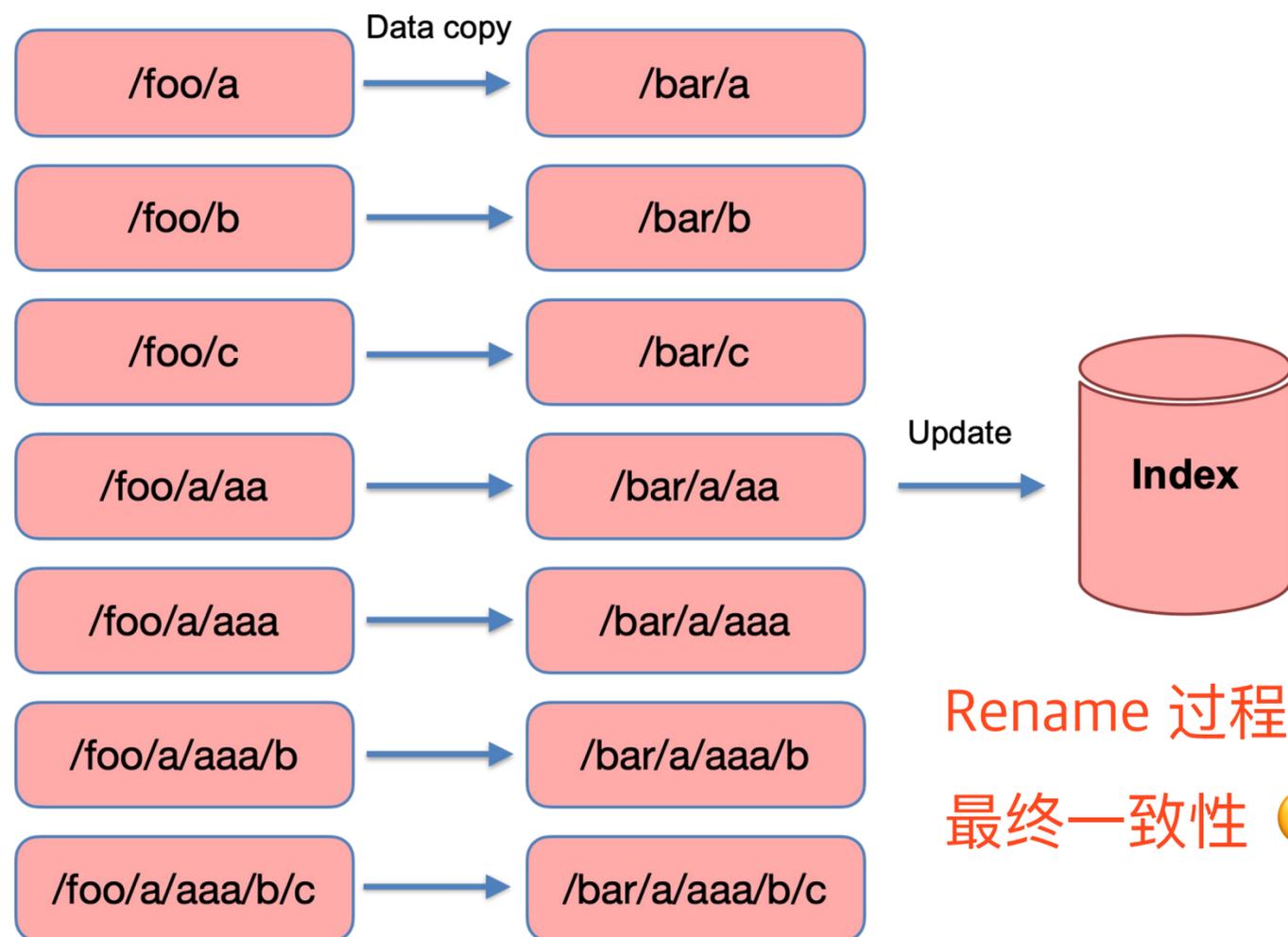
对象存储 vs. 文件系统

如何实现 Rename ? `> mv /foo /bar`

文件系统 📌



对象存储 📌

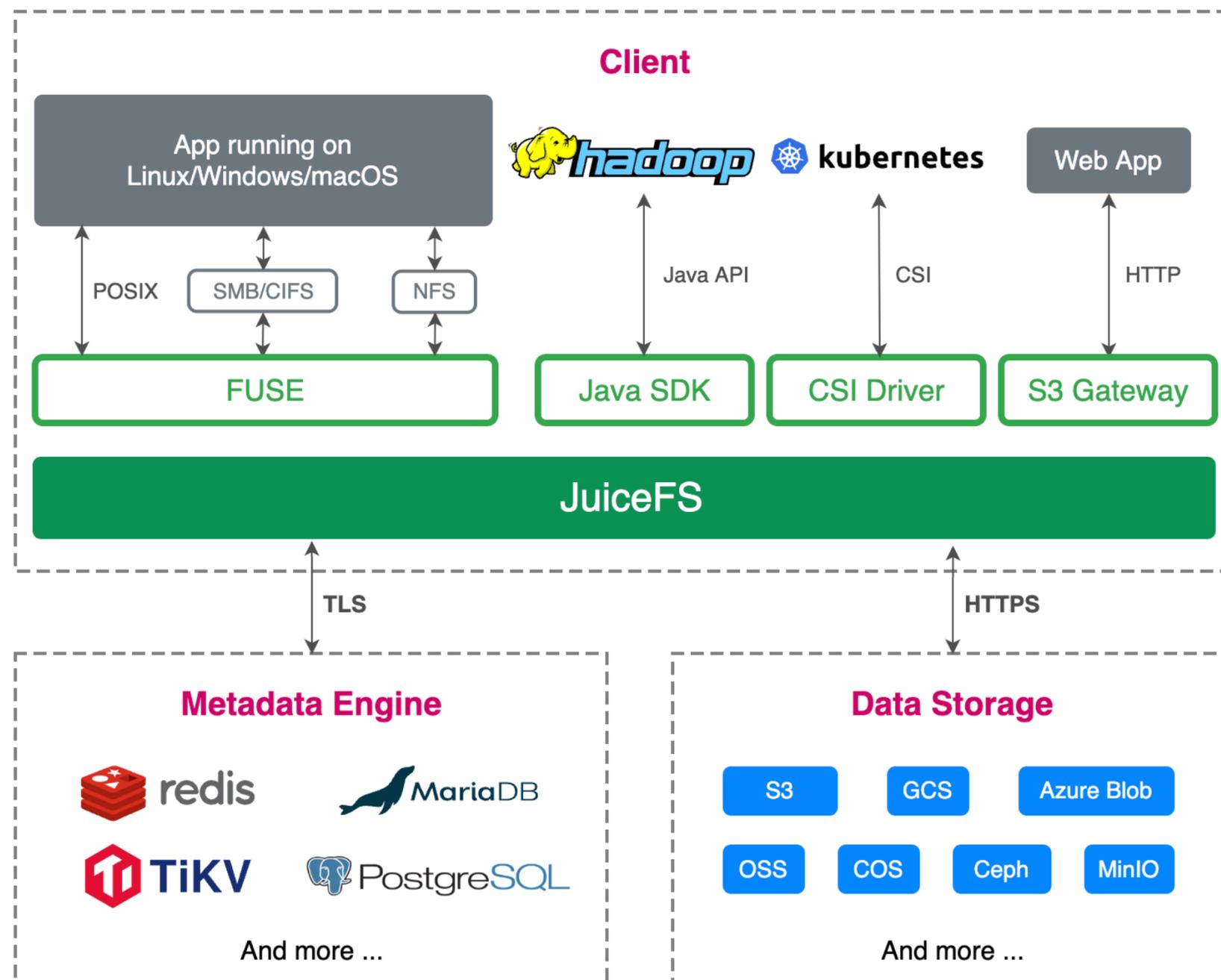


Rename 过程没有事务保证
最终一致性 😞

JuiceFS 的架构设计及原理解析

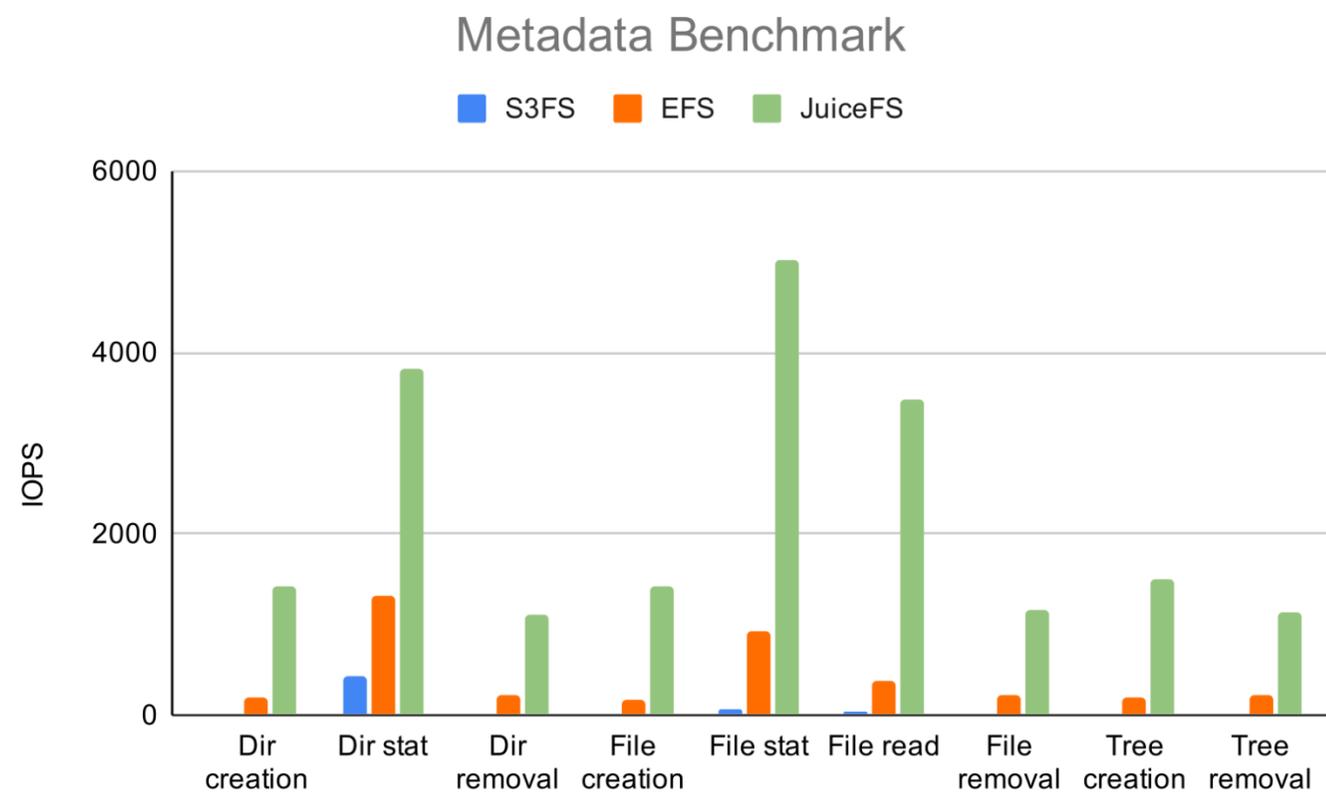
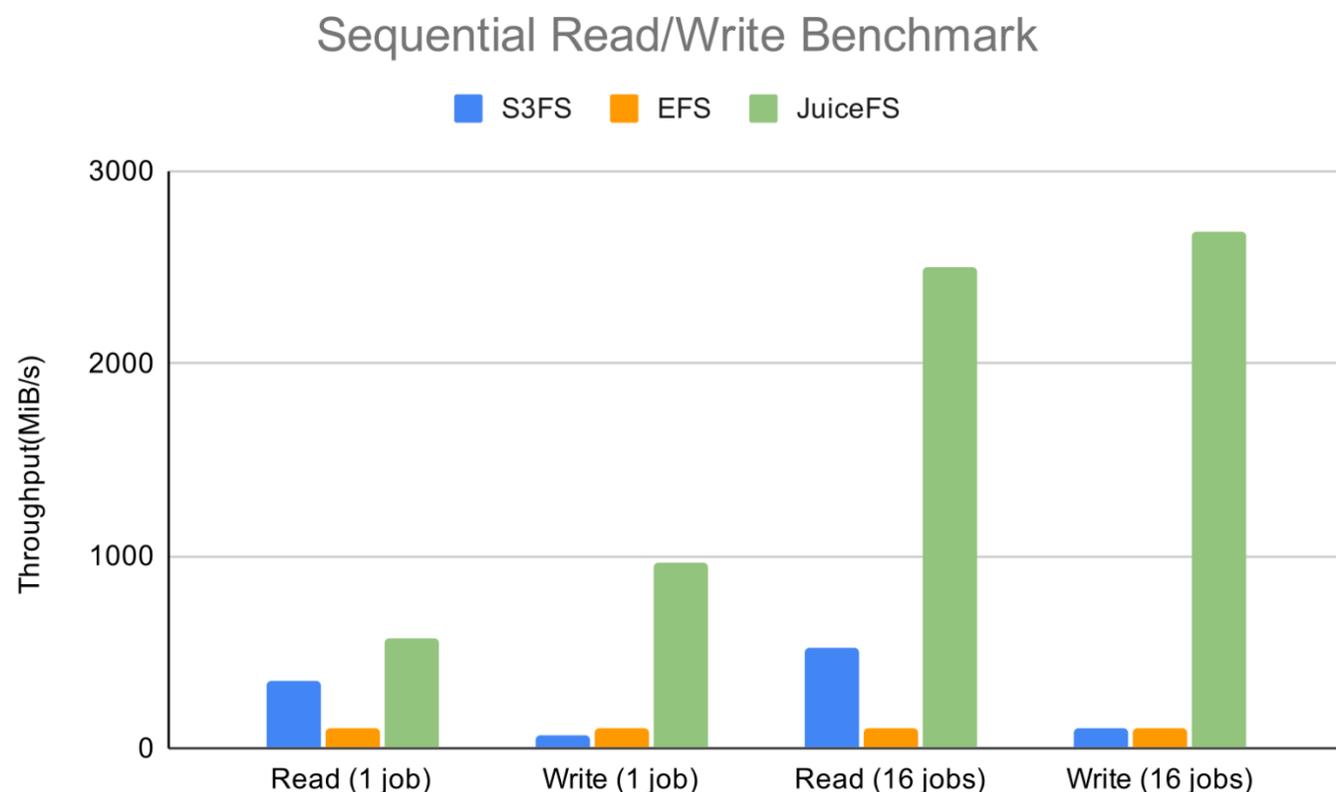
» JuiceFS 架构

- 强一致性分布式文件系统
- 插件式元数据引擎
- 使用对象存储作为数据存储
- 元数据引擎可横向扩展
- 小文件友好的元数据设计
- 本地多级缓存
- 多种类型客户端
- 兼容 POSIX、HDFS、S3 API





JuiceFS 读写性能



左图（越大越好）：使用 fio 进行顺序读写性能测试，对比 S3FS、Amazon EFS 和 JuiceFS 的读写吞吐。

右图（越大越好）：使用 mdtest 进行元数据性能测试，对比 S3FS、Amazon EFS 和 JuiceFS 的元数据请求 IOPS。

JuiceFS 在 Elasticsearch 的实践及案例

»» Why JuiceFS?

- 完全兼容 POSIX，应用无侵入。
- 弹性容量
- 存储成本远低于 SSD
- 写入性能远高于对象存储
- 本地缓存加速
- 运维便捷

» Elasticsearch x JuiceFS 冷热数据分层实践

- 准备多种类型节点，为不同节点分配不同的角色。
- Warm 或 Cold 节点挂载 JuiceFS 文件系统
- 创建生命周期策略
- 为索引设置生命周期策略（通过索引模板或 `index.lifecycle.name` 配置）

» Elasticsearch x JuiceFS 小提示

- Warm 或 Cold 节点的副本数 (replica) 可以设置为 0
- 开启 Force merge 可能会导致节点 CPU 持续占用，酌情关闭。
- Warm 或 Cold 阶段的索引可以设置为只读

Warm phase Move data into phase when: 30 days old

Move data to the warm tier, which is optimized for search performance over indexing performance. Data is infrequently added or updated in the warm phase.

Advanced settings

Replicas
Set the number of replicas. Remains the same as the previous phase by default.

Set replicas

Number of replicas: 1

Shrink
Shrink the index to a new index with fewer primary shards. [Learn more](#)

Shrink index

Force merge
Reduce the number of segments in each index shard and clean up deleted documents. [Learn more](#)

Force merge data

Number of segments: 1

Compress stored fields

Read only
Enable to make the index and index metadata read only, disable to allow writes and metadata changes. [Learn more](#)

Make index read only

Data allocation
Move data to nodes optimized for less-frequent, read-only access.

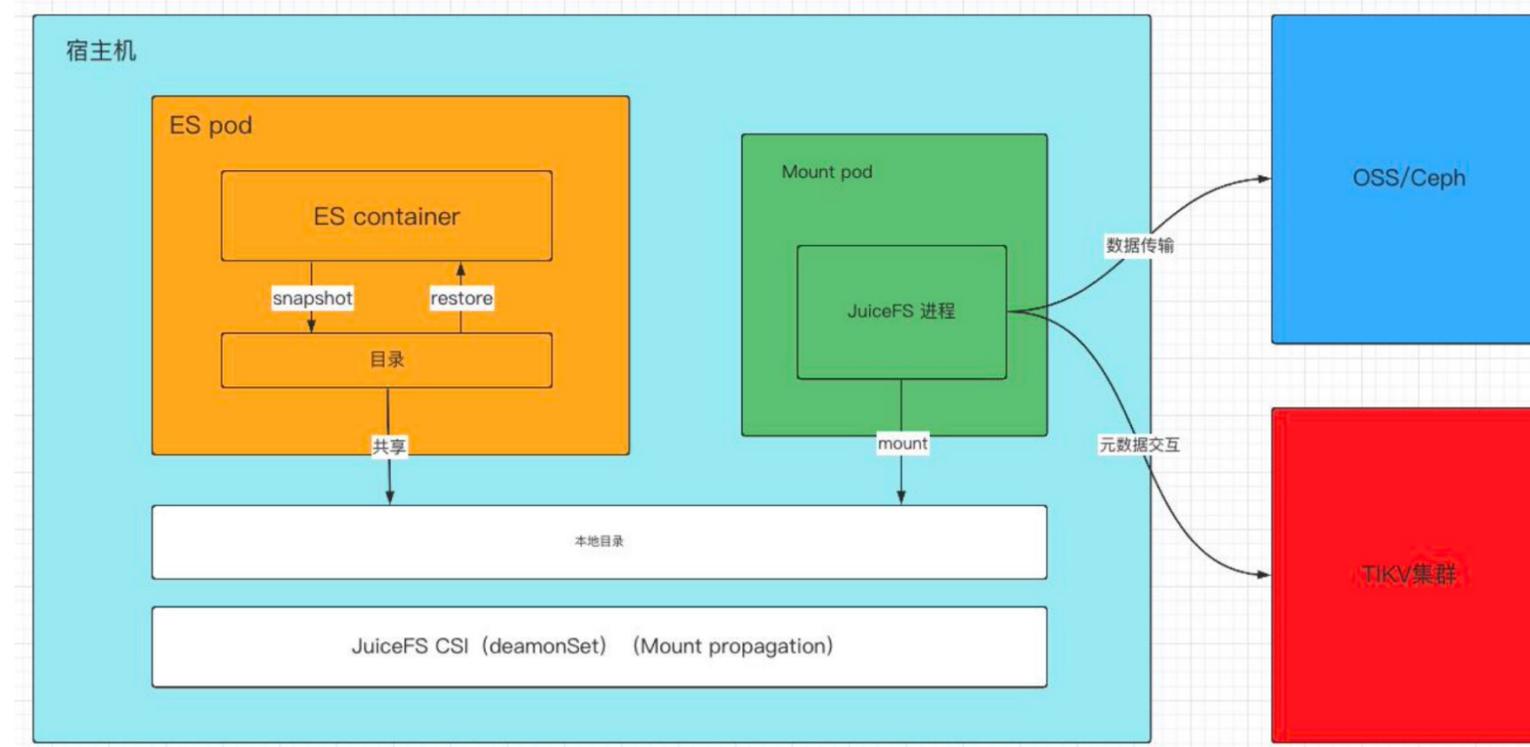
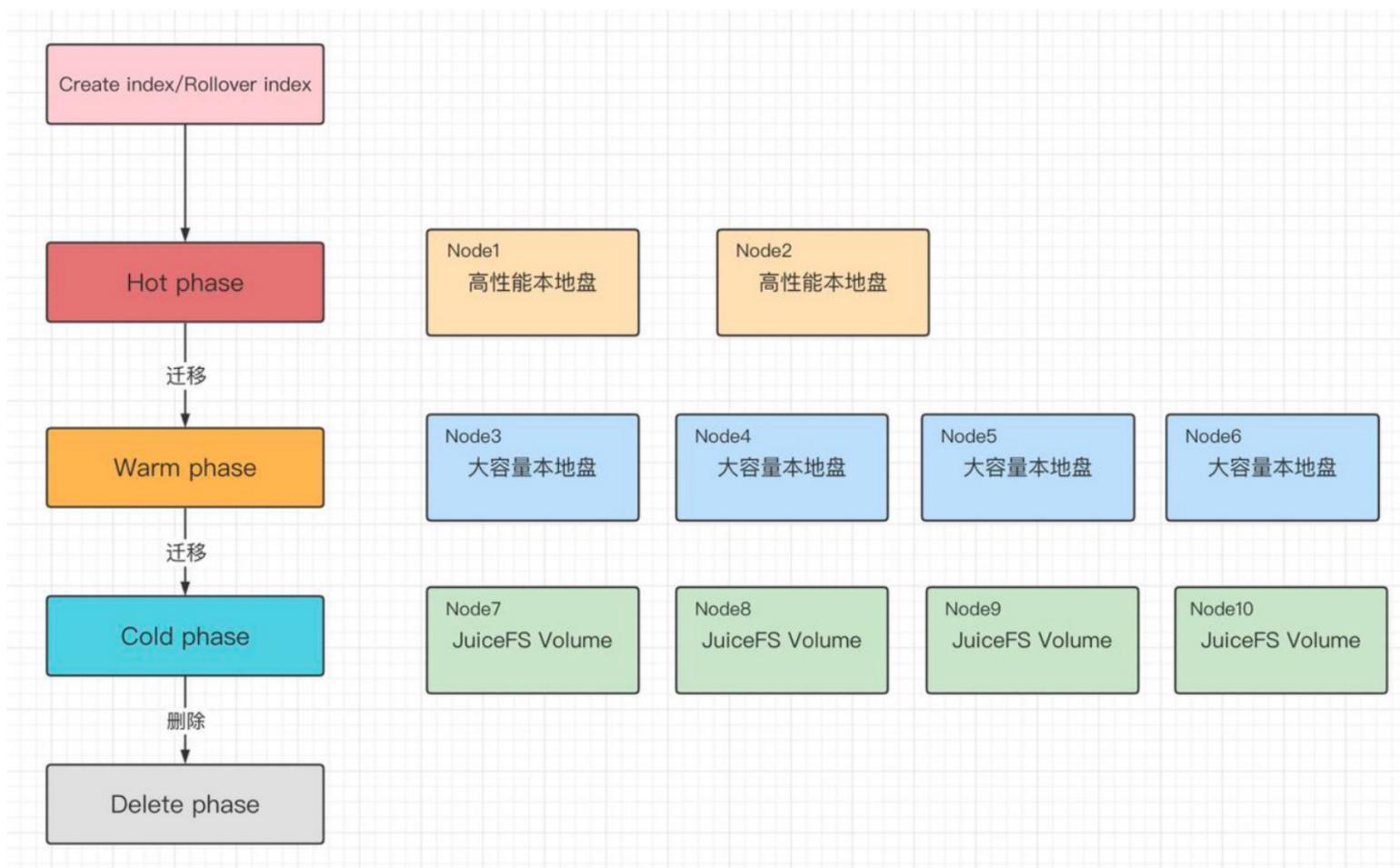
Data tier options: Use warm nodes (recommended)

Index priority
Set the priority for recovering your indices after a node restart. Indices with higher priorities are recovered before indices with lower priorities. [Learn more](#)

Set index priority

Index priority: 50

» Elasticsearch x JuiceFS 用户案例



携程已经在数据库备份和 Elasticsearch 冷数据存储场景对接了 JuiceFS，迁移了 2PB+ 的数据，预计后续还会有 10PB+ 的数据接入。





感谢观看



<https://github.com/juicedata/juicefs>



专业、垂直、纯粹的 Elastic 开源技术交流社区

<https://elasticsearch.cn/>