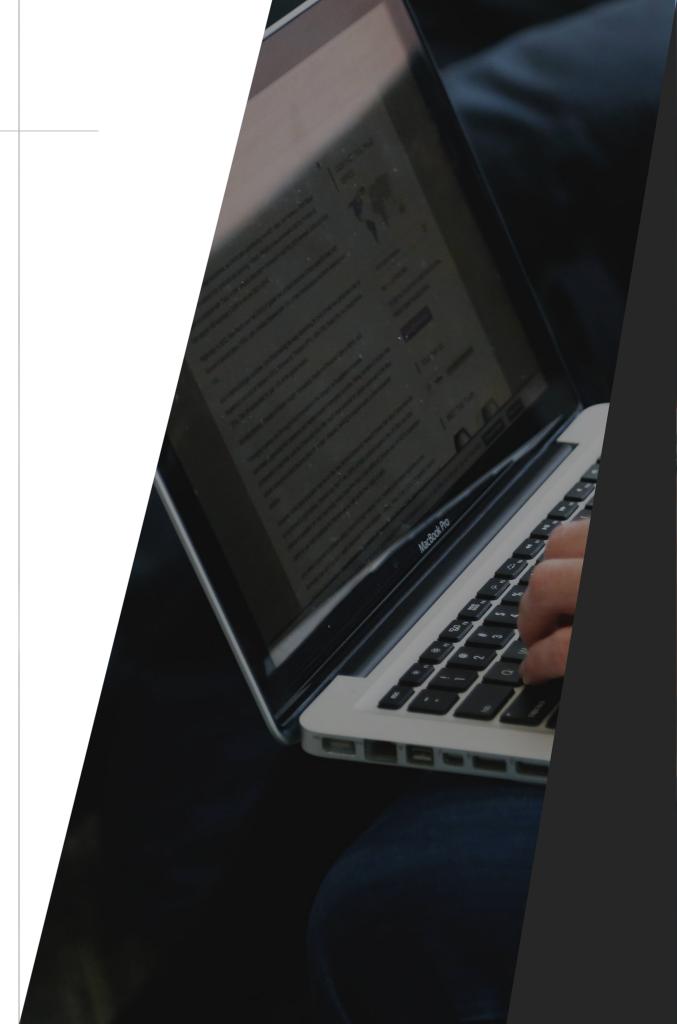


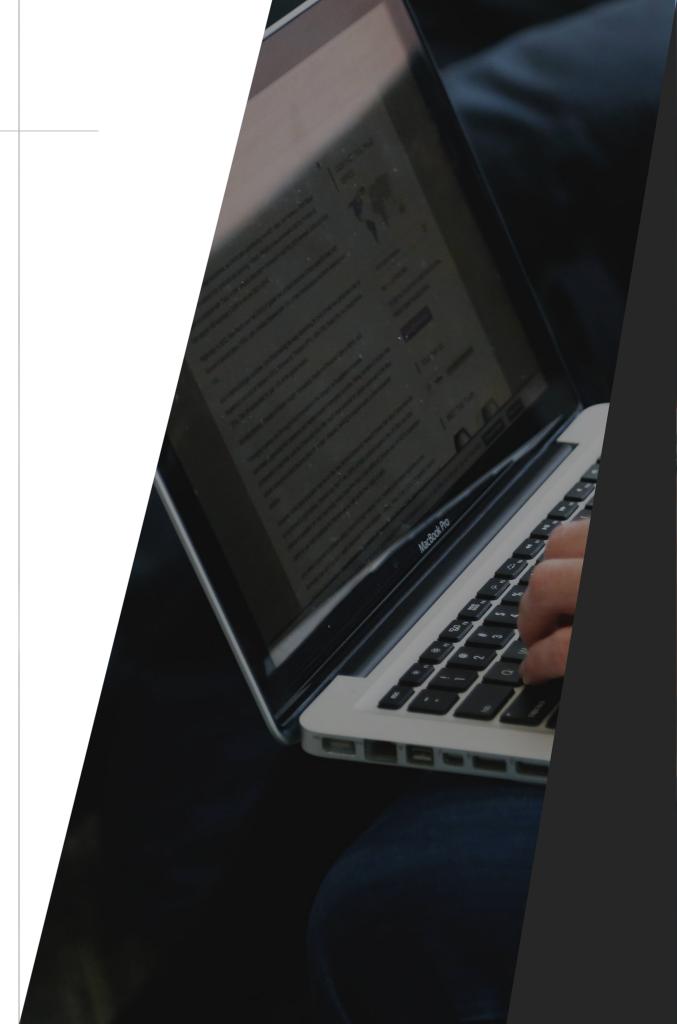
weiwan@sunteng.com



- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向



- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向

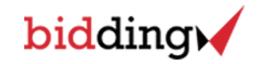


#### Why IndexR?

舜飞科技的实时分析需求:

- 单表300亿行,近百张表。
- 几十到上千字段。数字类型为主,少量字符串类型。
- · 96%查询响应5s以内,常用高频查询1s。
- 实时入库+离线导入。
- · 多维分析为主,包括复杂SQL。无法预计算。
- 100%精确,不能丢失数据,或者使用近似查询。
- 增删字段不影响业务。

IndexR集群11个节点,200T数据,多条业务线:



DSP程序化购买



网站检测分析



AB测试



#### What is IndexR?

不仅是极快的On Hadoop数据存储格式,还是专注于OLAP的数据仓库

- 速度快
- 自带索引
- 支持实时导入
- 支持预聚合
- 支持在线更新表结构
- 成本低
- etc.



#### How we use IndexR?

· 和Parquet一样,用作大数据的存储格式。目前支持Spark、Drill以及Hive。

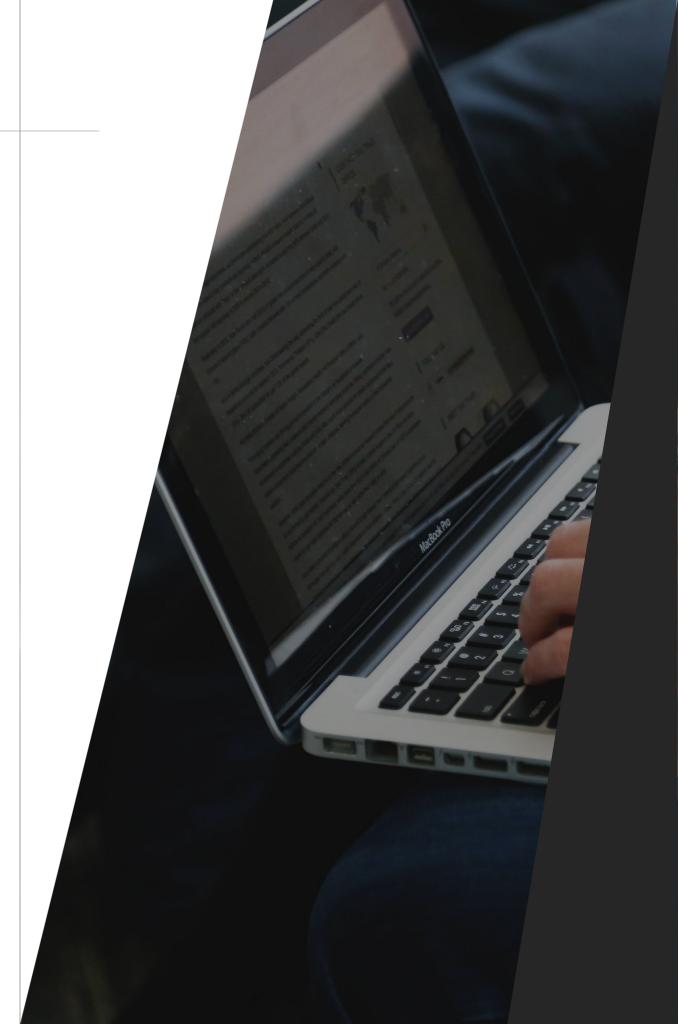
2x~8x Query Speed up

·与Hadoop生态的其他模块配合,组成一个数据仓库。存放海量历史数据, 支持在线交互式分析查询(OLAP)

IndexR + Hadoop Ecosystem = Data Warehouse with OLAP



- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向



## 数据仓库架构

Drill / Spark

Distributed Realtime Query Engine Kafka

Realtime Ingestion

IndexR

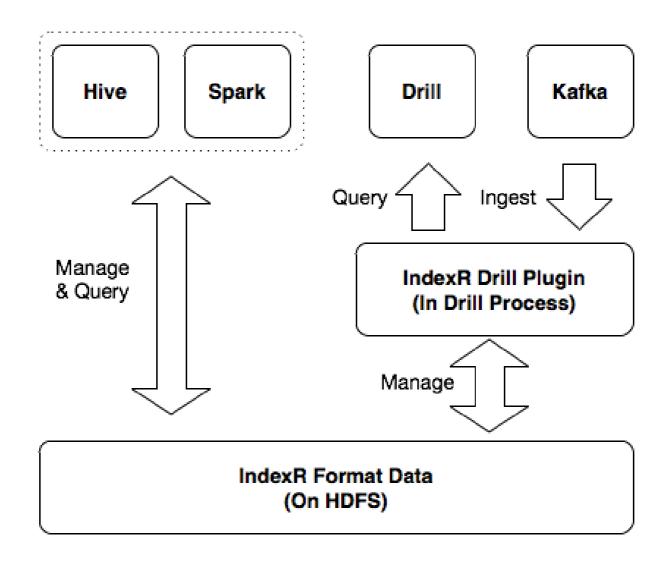
Storage Format, Realtime Data Management, Query optimization, Cache

Hadoop Ecosystem

Distributed Storage, Offline Computing, Offline Data Management, ETL Toolset(Hive)

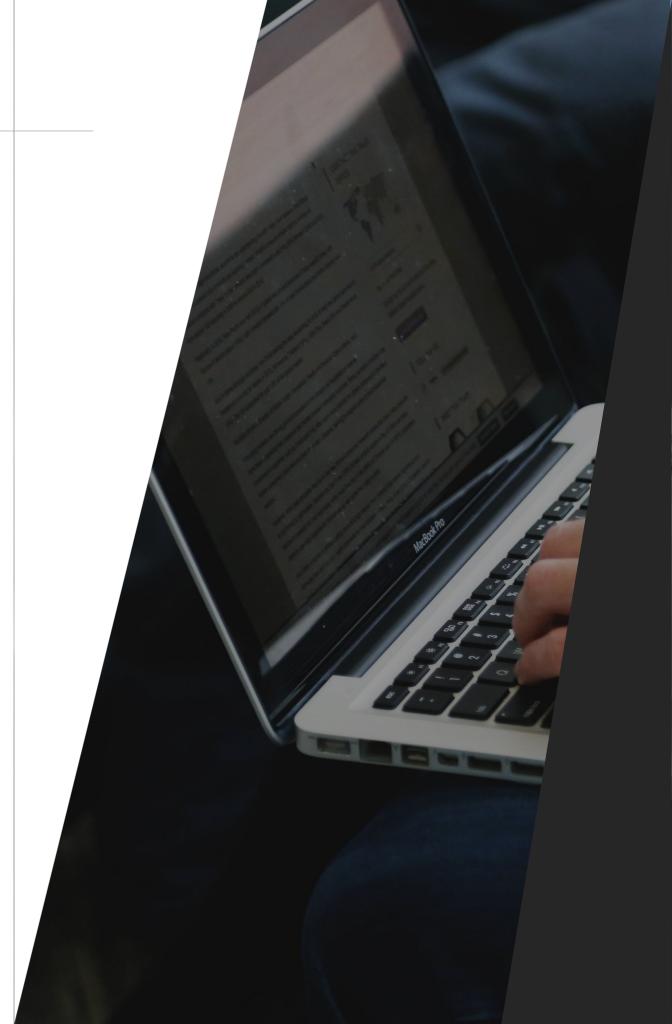
# Zookeeper

## 查询引擎对IndexR的使用方式

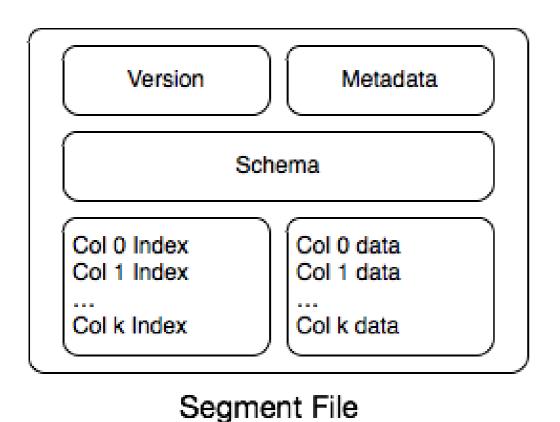




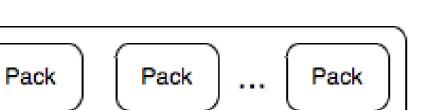
- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向



## 文件格式



- · Segment文件组成表
- 列式存储
- 自解释
- 自带索引



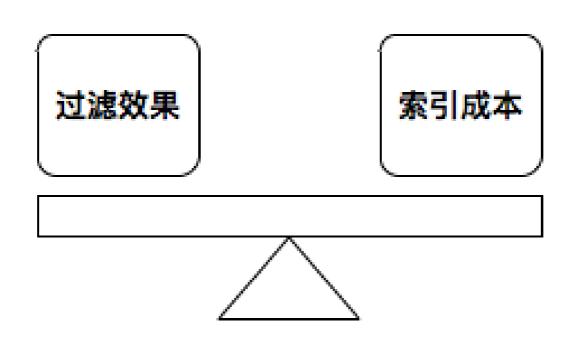
Column

- · 列数据被划分成多个Pack
- · 一个Pack包含65536个数据
- · Pack是基本的IO和索引单位



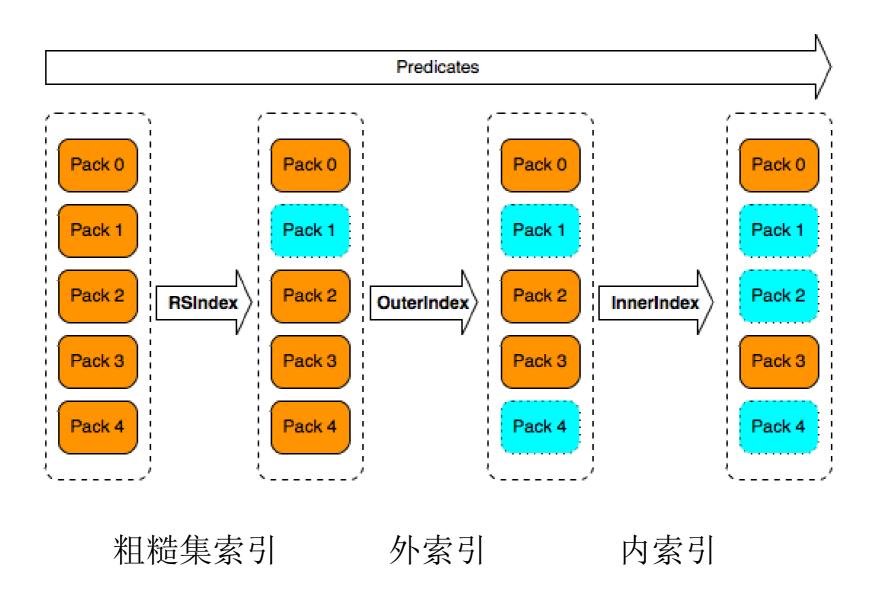
## 索引

索引对于大数据的实时分析是一把双刃剑, 关键在于平衡与取舍。





## 三层索引

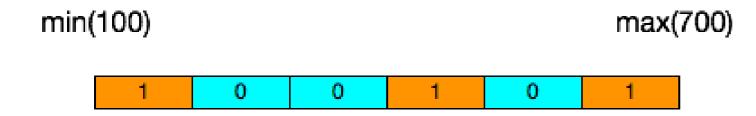




#### RSIndex-数字类型



· 从最简单直观的min-max开始



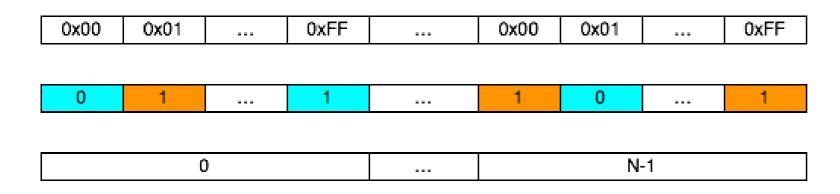
- 在有效范围之内, 等距离切割
- · 每一个小块(block)用一个bit表示
- · 对于block,如果Pack有数据在它区间之内,则为1,否则为0

a = 230	Not match
a = 450	Match

• 过滤示例



#### RSIndex-字符串类型



- · 字符串看做byte数组
- · byte有256个值(slot)
- · 某个slot为1表示:存在字符串S,它的第k位byte值为该slot

#### N-len String Index

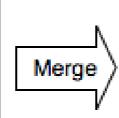
0-len index 1-len index		64-len index
-------------------------	--	--------------

· 进一步,把Pack内的字符串按照长度分组,分别索引

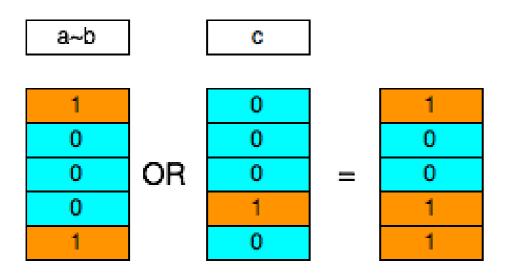


#### **OuterIndex**

	a	b	С	d
pack0	0	1	0	0
pack1	1	0	0	0
pack2	0	1	0	0
pack3	0	0	1	0
pack4	1	0	0	1



	a~b	<b>~</b> d
pack0	1	0
pack1	1	0
pack2	1	0
pack3	0	1
pack4	1	1



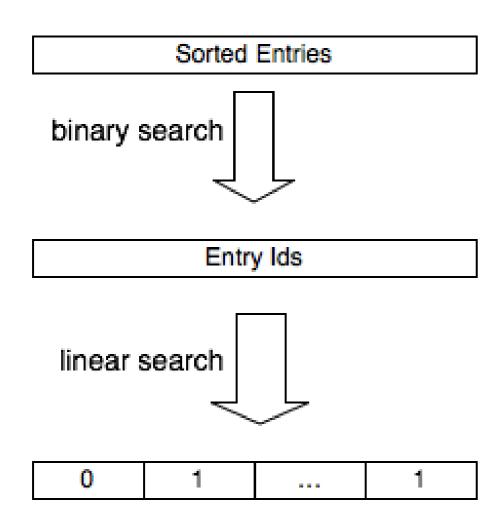
- 倒排索引 (Inverted Index)
- · 索引pack, 而非行
- 使用bitmap存储entry到pack的映射
- · 1表示pack中存在该值
- · entry按照范围分割成block,额外存储block之内merge的bitmap

col between 'a' and 'c'

思考:如果是 not(predicate) 怎么过滤?



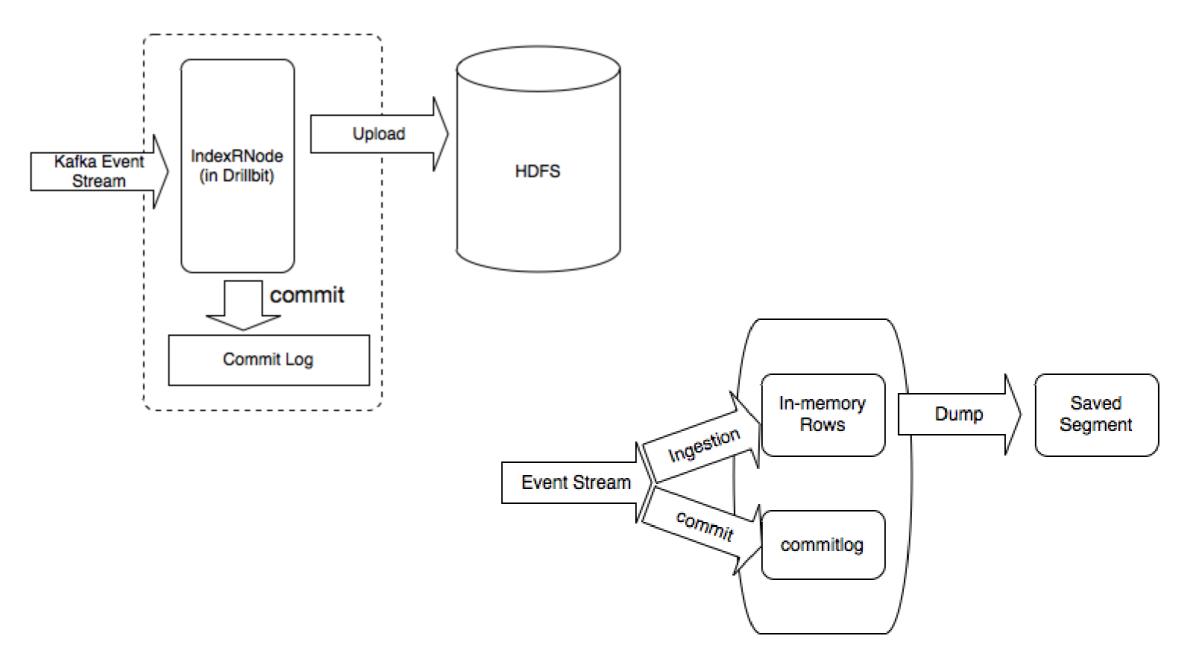
#### InnerIndex



- 排序有利于压缩
- · 利用二分搜索,快速确定有效pack
- · int运算比字符串快
- 利用bitmap



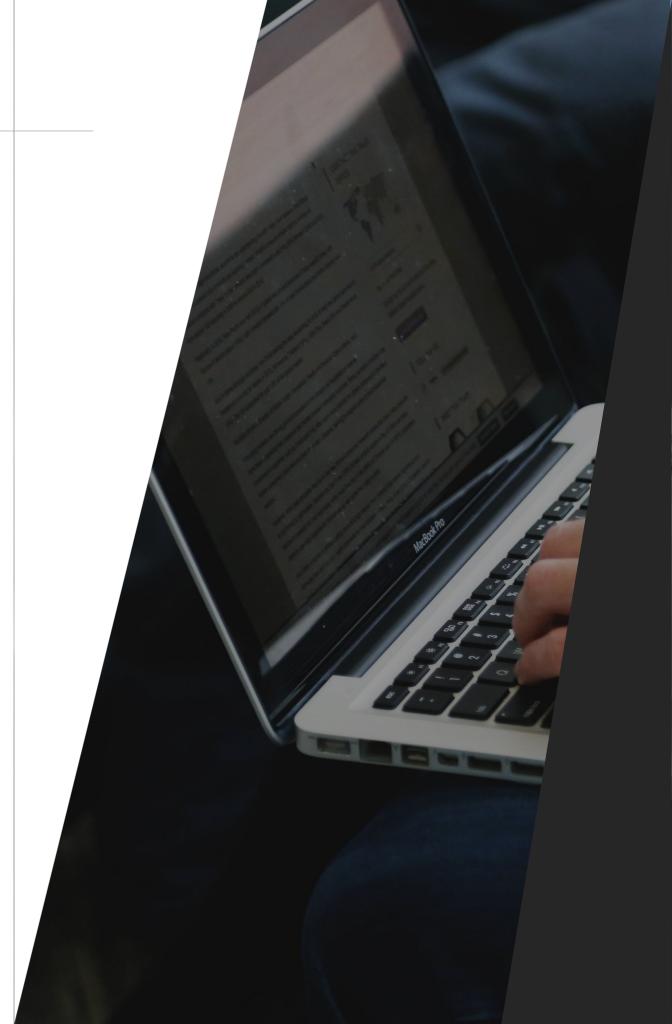
## 实时入库



Ingesting Realtime Segment



- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向

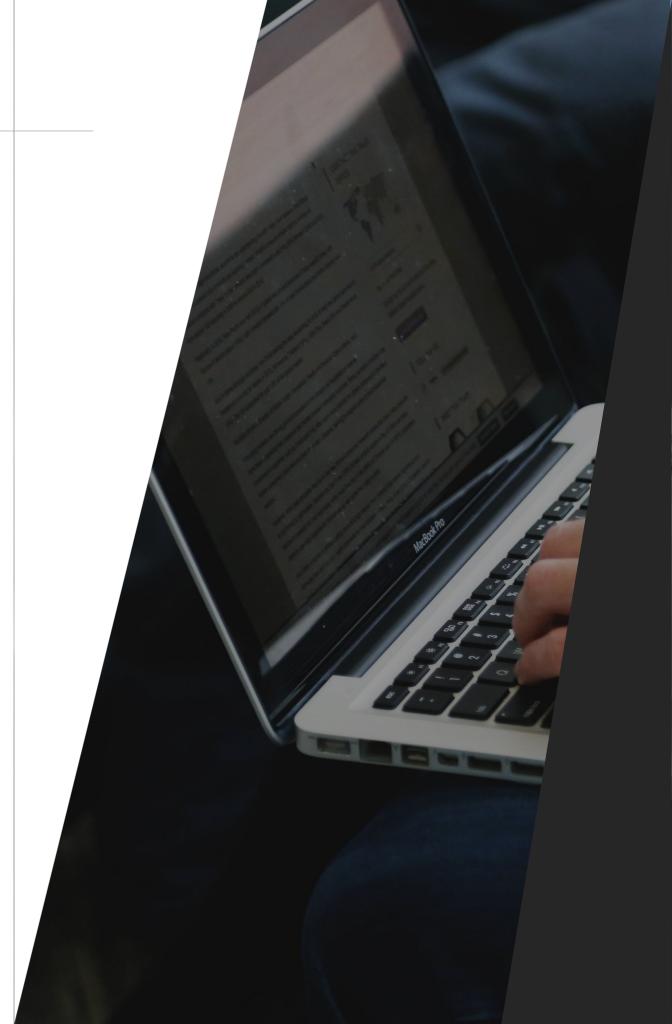


#### 代码优化

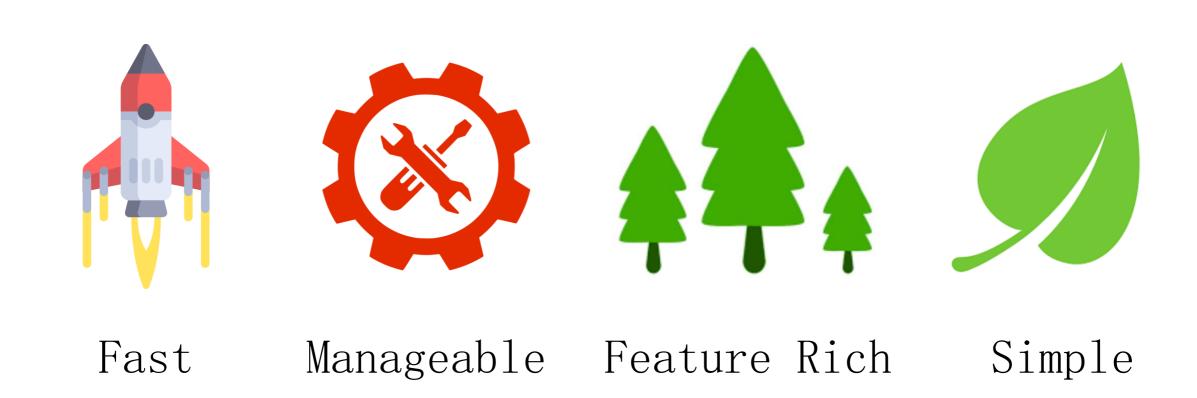
- Use Off-Heap Memory
- Use Vector Processing
- Implement Java Like C (could help)
- Avoid Random IO
- No SerDes
- Measure, Don't Guess!



- **01** IndexR的定位
- 02 架构和生态
- 03 技术特点
- 04 设计理念
- 05 发展方向



#### IndexR的发展方向



IndexR立志成为Hadoop生态内实时分析的流行工具



#### Info

- <a href="https://github.com/shunfei/indexr">https://github.com/shunfei/indexr</a>
- weiwan@sunteng.com
- WeChat: xilyflow

