

# Twitter实时搜索引擎 进化历程

# Geekbang>

## 极客邦科技

整合全球最优质学习资源, 帮助技术人和企业成长  
Growing Technicians, Growing Companies

**InfoQ**

专注中高端技术人员的  
技术媒体



**EGG** EXTRA GEEKS' ORGANIZATION  
NETWORKS

高端技术人员  
学习型社交网络



**StuQ**

实践驱动的  
IT职业学习和服务平台



**GiT** GEEKBANG  
INTERNATIONAL  
TRAINING

一线专家驱动的  
企业培训服务



旧金山 伦敦 北京 圣保罗 东京 纽约 上海  
San Francisco London Beijing Sao Paulo Tokyo New York Shanghai

# QCon

## 全球软件开发大会

2016年4月21-23日 | 北京·国际会议中心

主办方 **Geekbang** & **InfoQ**  
极客邦科技

**7折** 优惠 (截至12月27日)  
现在报名, 节省2040元/张, 团购享受更多优惠

[www.qconbeijing.com](http://www.qconbeijing.com)



扫描获取更多大会信息

## 个人介绍

# 王天

2003年本科毕业于清华大学计算机系；

2005年硕士毕业于University of North Carolina Chapel Hill；

2005年7月加入Google，从事移动搜索、新闻搜索、搜索质量等工作；

2011年3月加入Twitter搜索部门，工作至今。

带领Twitter的搜索质量团队，改进实时搜索产品。



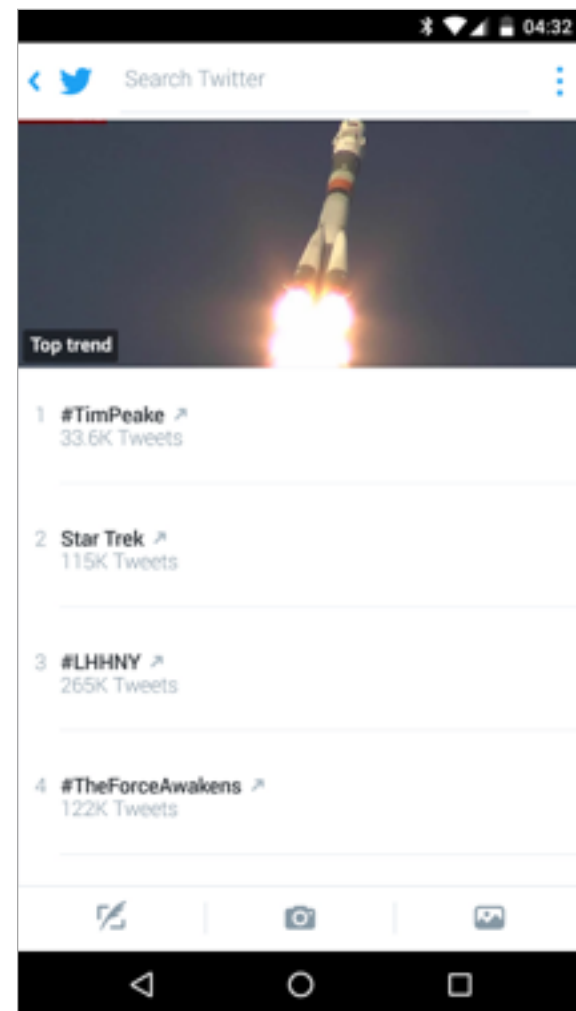
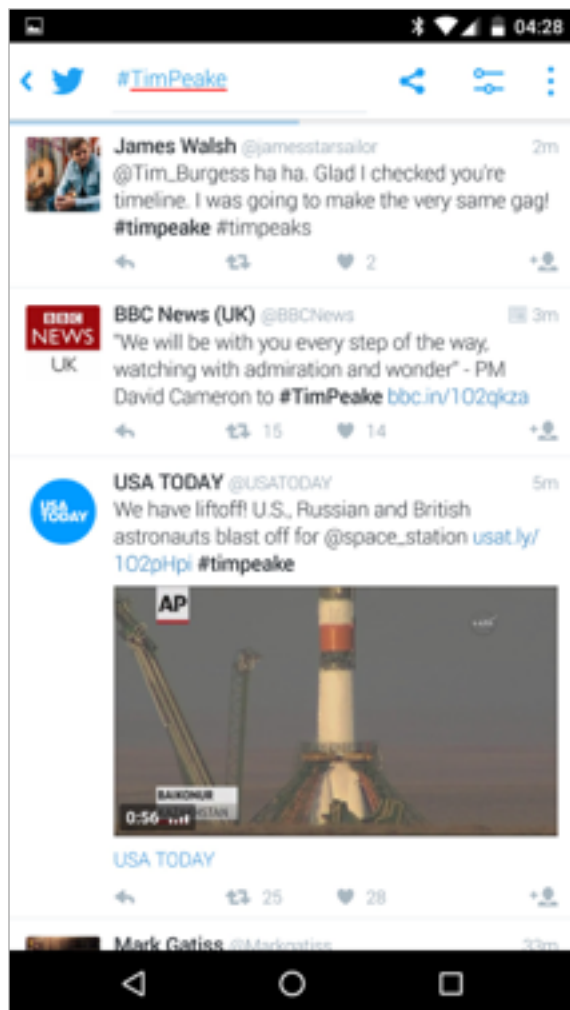
@wangtian

[www.twitter.com/wangtian](https://www.twitter.com/wangtian) 

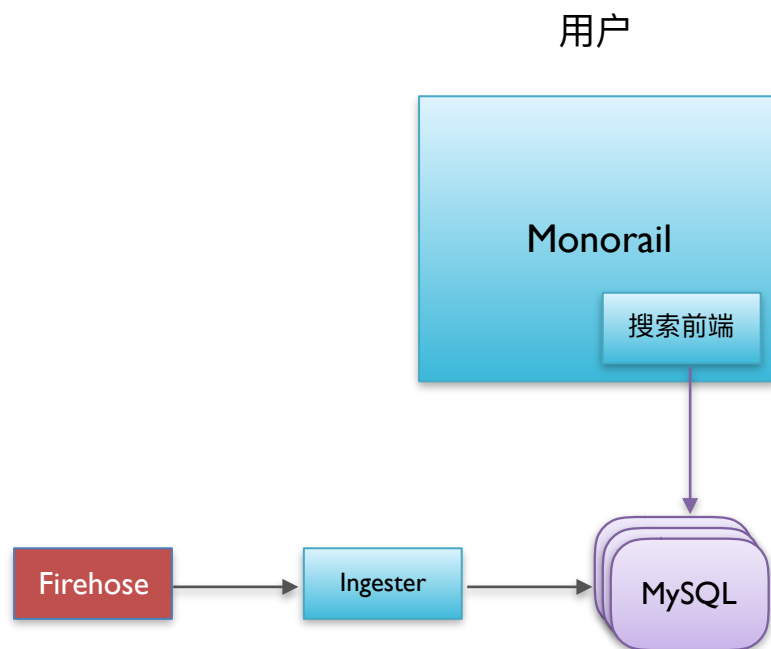
[www.linkedin.com/in/wangtian](https://www.linkedin.com/in/wangtian) 

## Twitter搜索概况

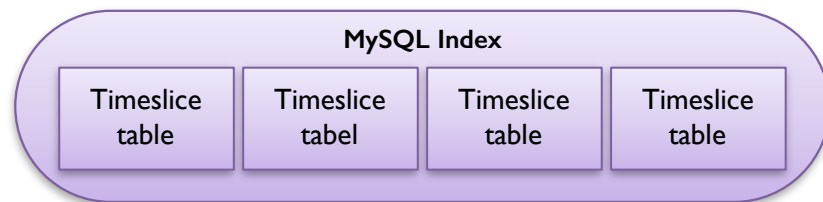
- 2008年始于收购的Summize公司
- 核心服务：关键字搜索、相关性搜索、Facets搜索、Term统计...
- 相关服务：自动建议、拼写检查、相关推荐...
- 6700亿条推文全部纳入索引
- 来自Twitter搜索产品
  - Web、Android、iOS、TweetDeck、...
- 来自公共、合作伙伴及内部API
  - 其他开发者的程序、网站、移动应用、桌面应用
  - 商业搜索服务订阅者（如GNIP）



2008 ~ 2009



- 推文被直接写到数据库
- 写满一个表就开启下一个
- 搜索直接在数据库进行
- 只支持三天左右的数据



基于数据库的索引

# 问题

## 性能

- 容量有限，扩展困难
- 搜索并发性低
- 实时程度受数据库索引更新影响

## 功能前景

- 难以添加新数据
- 难以支持复杂的查询语法
- 无法支持相关性搜索
- 搜索前端存在于更大系统中，难以开发



## 基于Lucene的索引重写

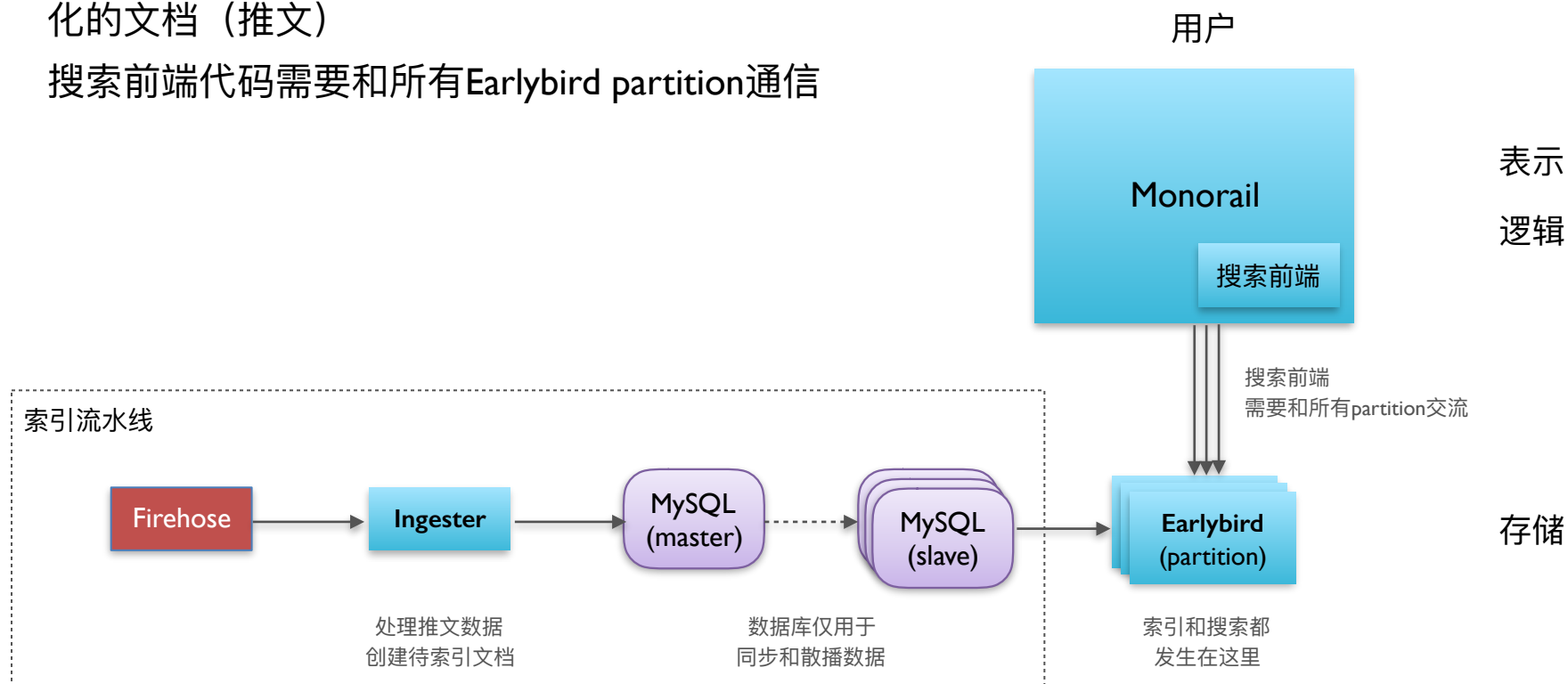
- API兼容情况下重写存储层posting list结构
  - 实时更新、多线程无锁读访问
  - 优先支持逆时间序的搜索
  - 支持提前结束搜索
  - 全内存索引
  - 支持地理数据
- 新树状可扩展query结构
  - Lucene query上面独立包装的数据结构
  - 规范query元素
- 从time partition到hash partition
- 速度提升：5ms中值延迟
- 服务容量提升：10x
- 反向贡献至Lucene开源社区

# 查询语言：Search Query

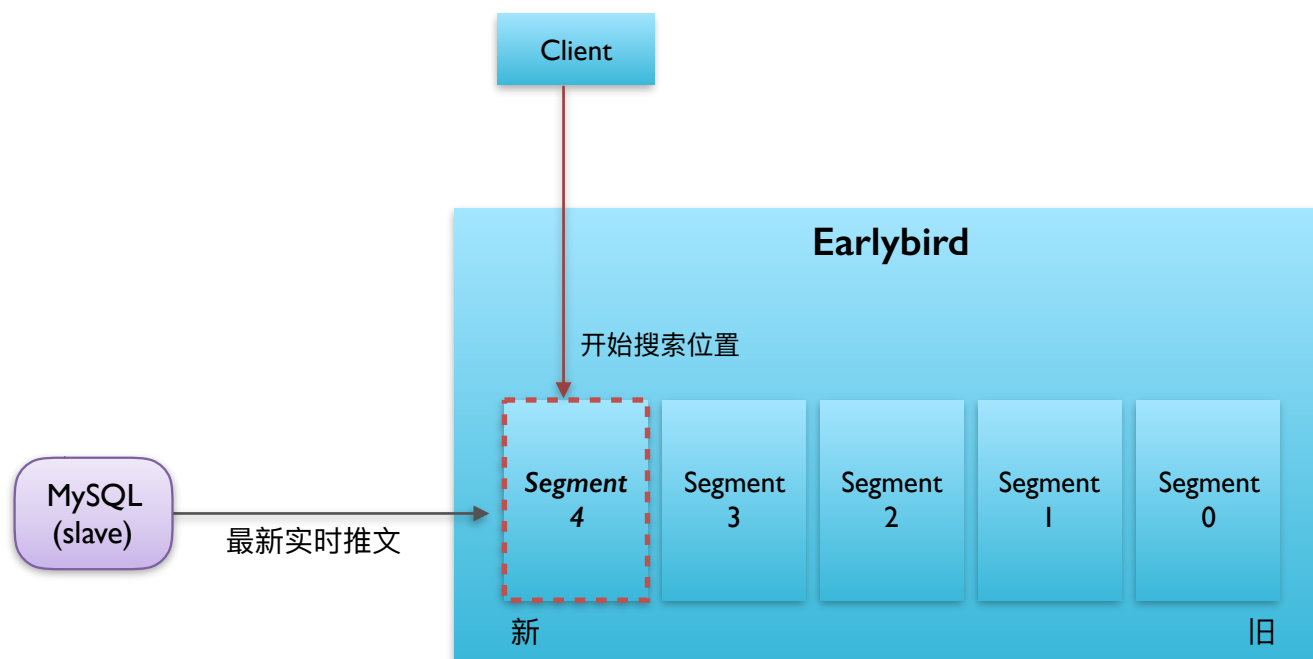


## 新一代索引服务器：Earlybird (2010)

- Earlybird既是一个索引器(Indexer)也是一个索引服务器(Index Server)
- MySQL现在仅用于同步和散播数据，保存序列化的文档（推文）
- 搜索前端代码需要和所有Earlybird partition通信



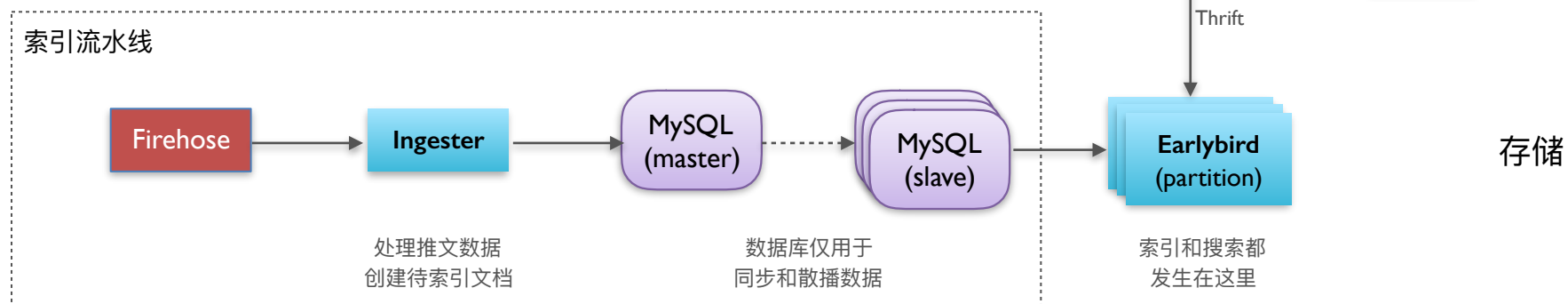
# Earlybird



**Segment:** 一个物理独立的Lucene Index  
Posting List中documents按照时间逆序排列

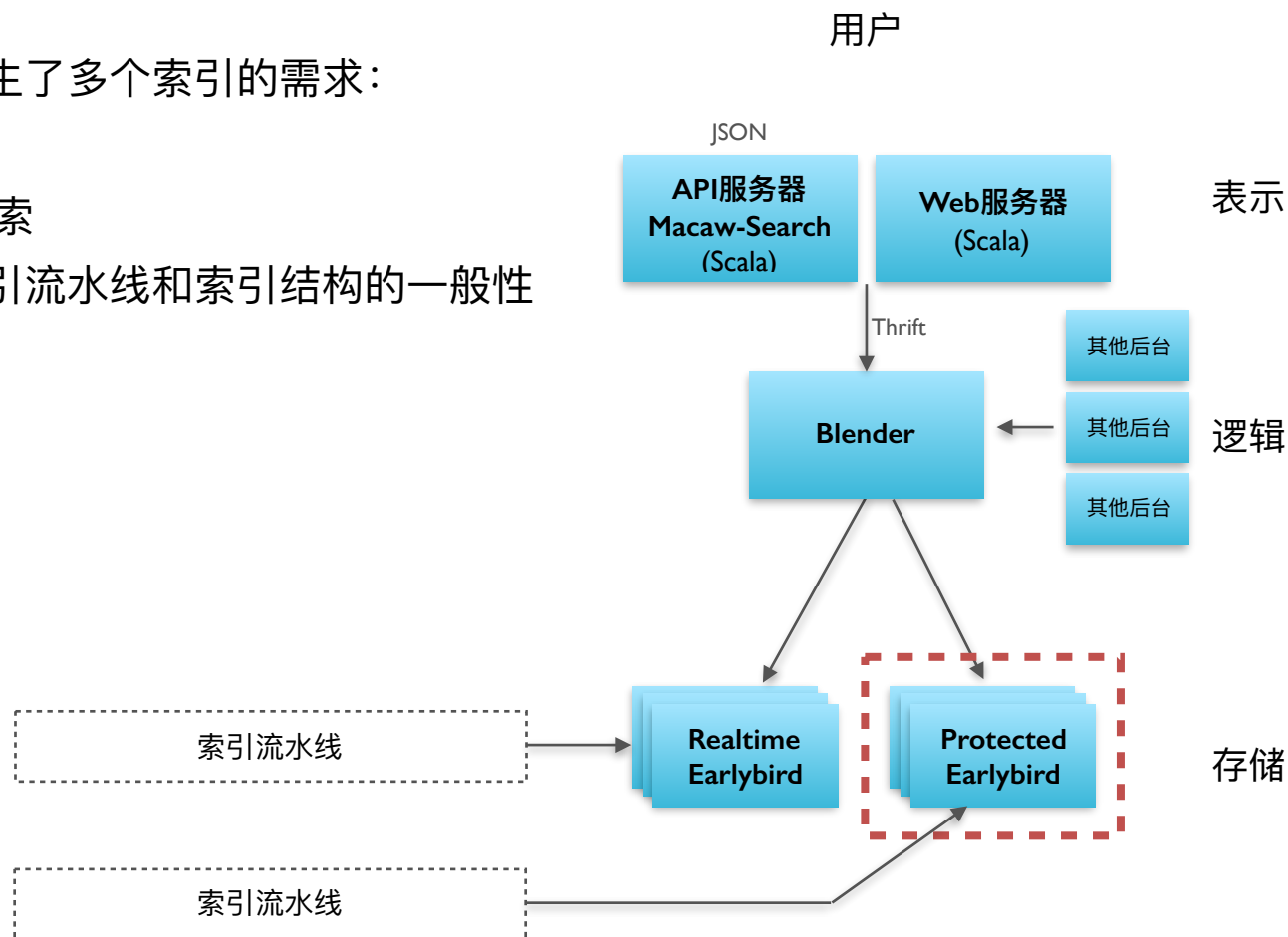
## 新一代搜索服务器：Blender (2011)

- 一开始作为一个中间层的Earlybird根节点出现
- 附带进行其它附加操作：过滤、填充结果、...
- 设计初期即考虑了通用性，支持多workflow\*
- 各种运营便利：
  - 彻底解耦表示和逻辑
  - 对于后台service的统一包装
  - 统一错误处理和跟踪



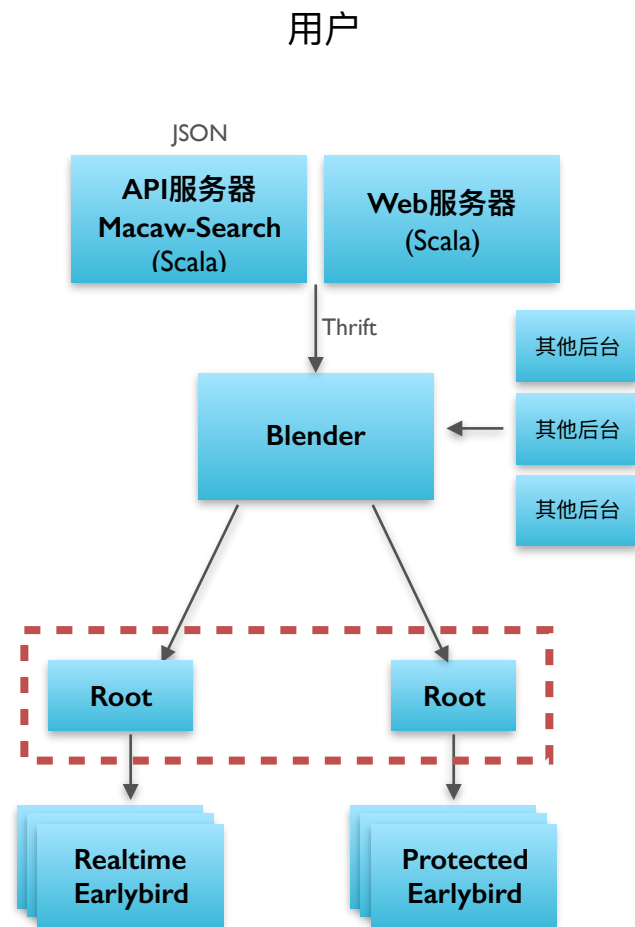
## 新索引服务器cluster出现

- 因为产品原因产生了多个索引的需求：
  - 实时搜索
  - 隐藏推文搜索
- 需要重新审视索引流水线和索引结构的一般性



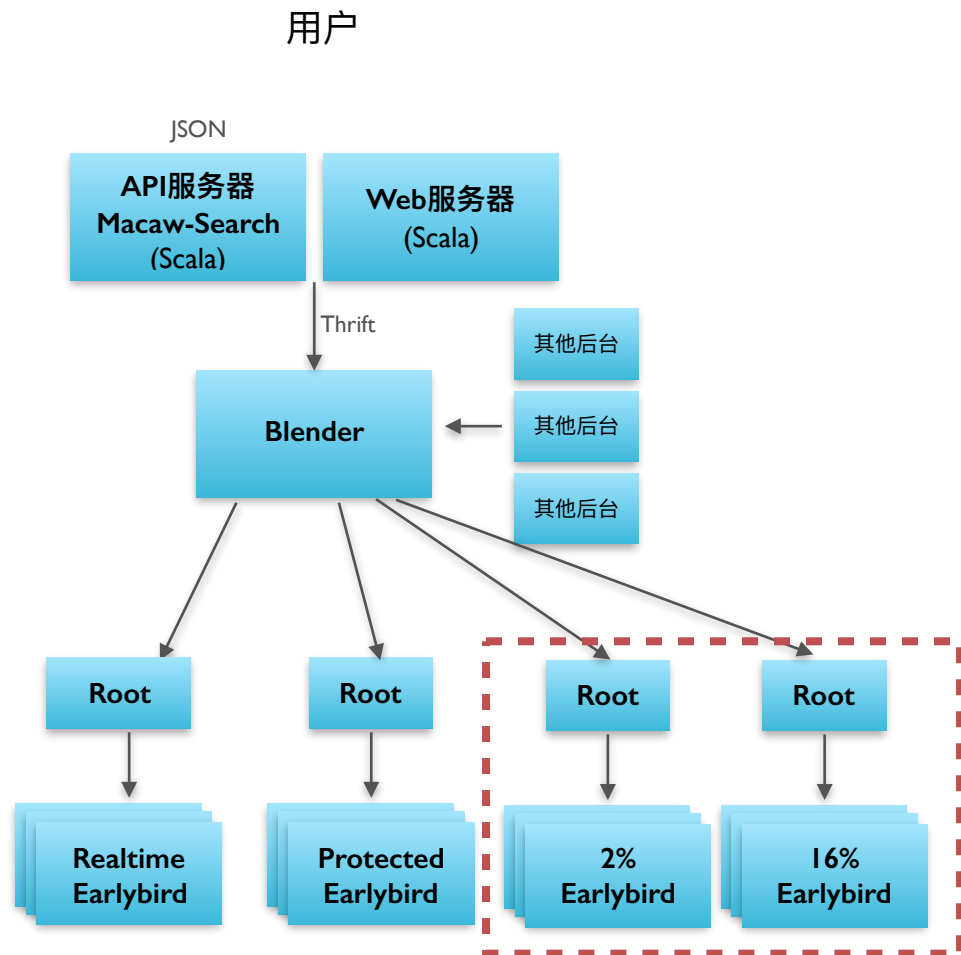
## 视图简化（一）：Root

- 提供partition无关的索引视图
- 增加轻量级索引根节点Root
- 隐藏索引cluster拓扑信息，统一处理结果合并问题



## 走向完全索引（一）

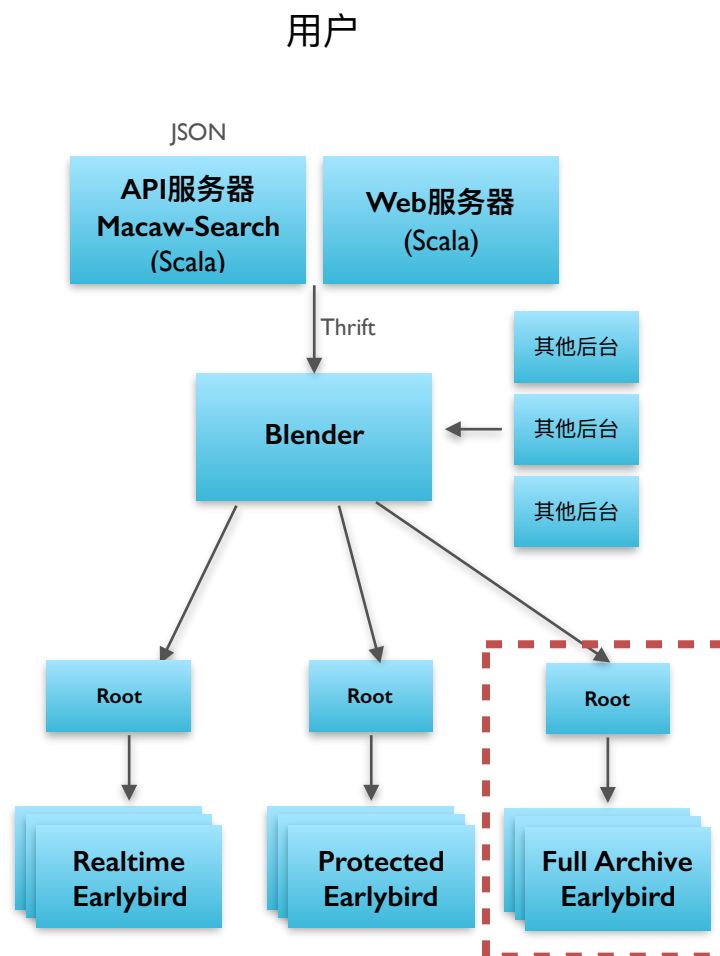
- 我们仍然只能搜索一周左右的推文
- Twitter从2006到现在累积了6700亿条历史推文
- 需要完整索引，但是按照Realtime索引方式实现代价不菲
- 计划分多阶段考察早期推文的搜索实用性
  - 2% Top (in-memory serve)
  - 16% Top (SSD)
  - Full Index (SSD)
    - 可扩展多层实现
    - 顺序搜索





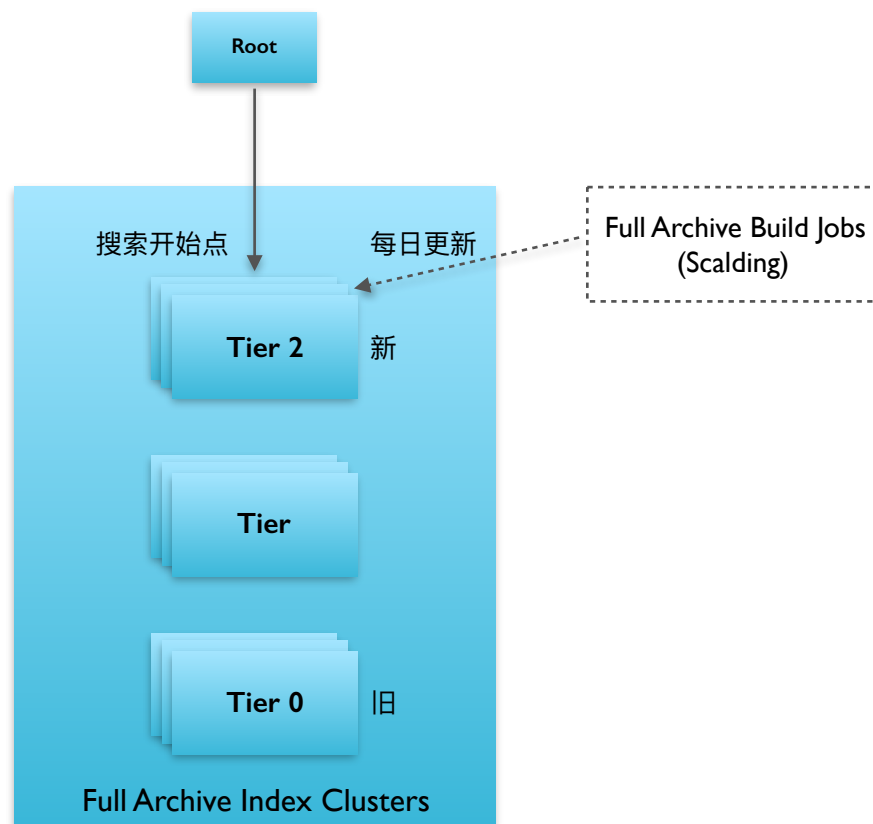
## 走向完全索引（二）

- 经过一年多的渐进实现，我们逐渐了解了完全索引的存在对用户行为的影响
- 实现了可垂直扩张的完全索引cluster



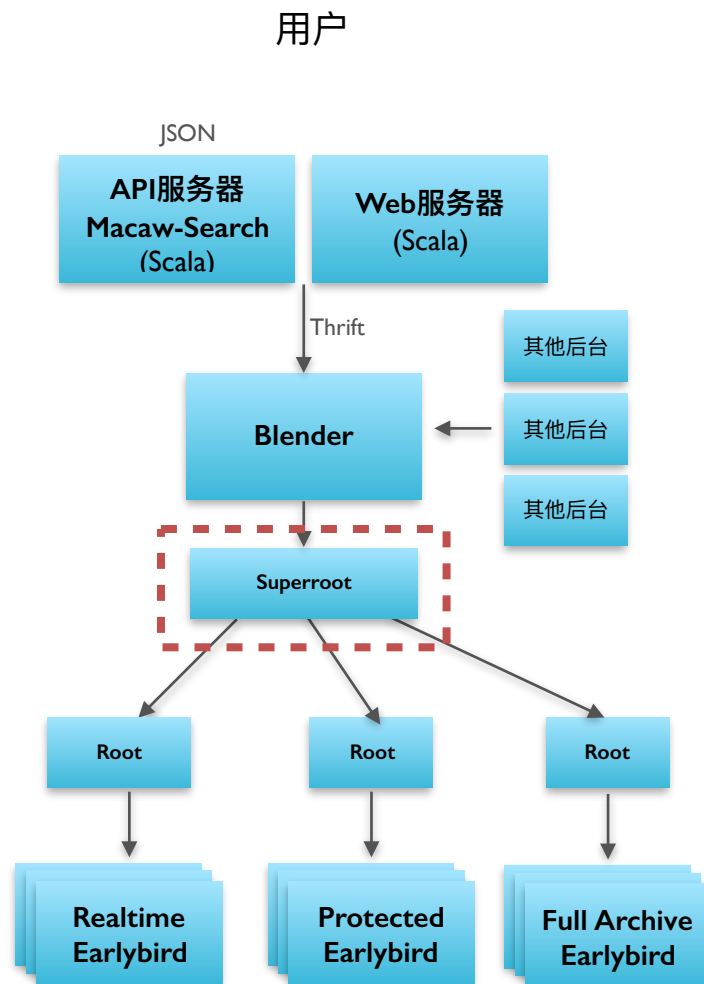
## 完全索引: Full Archive Index

- 按时间划分的多层索引
- 顶层索引每日更新（3天延迟）
- 一个tier装满后在上面建立新的tier
- 搜索总是从最新的tier开始
- 按用户ID来partition

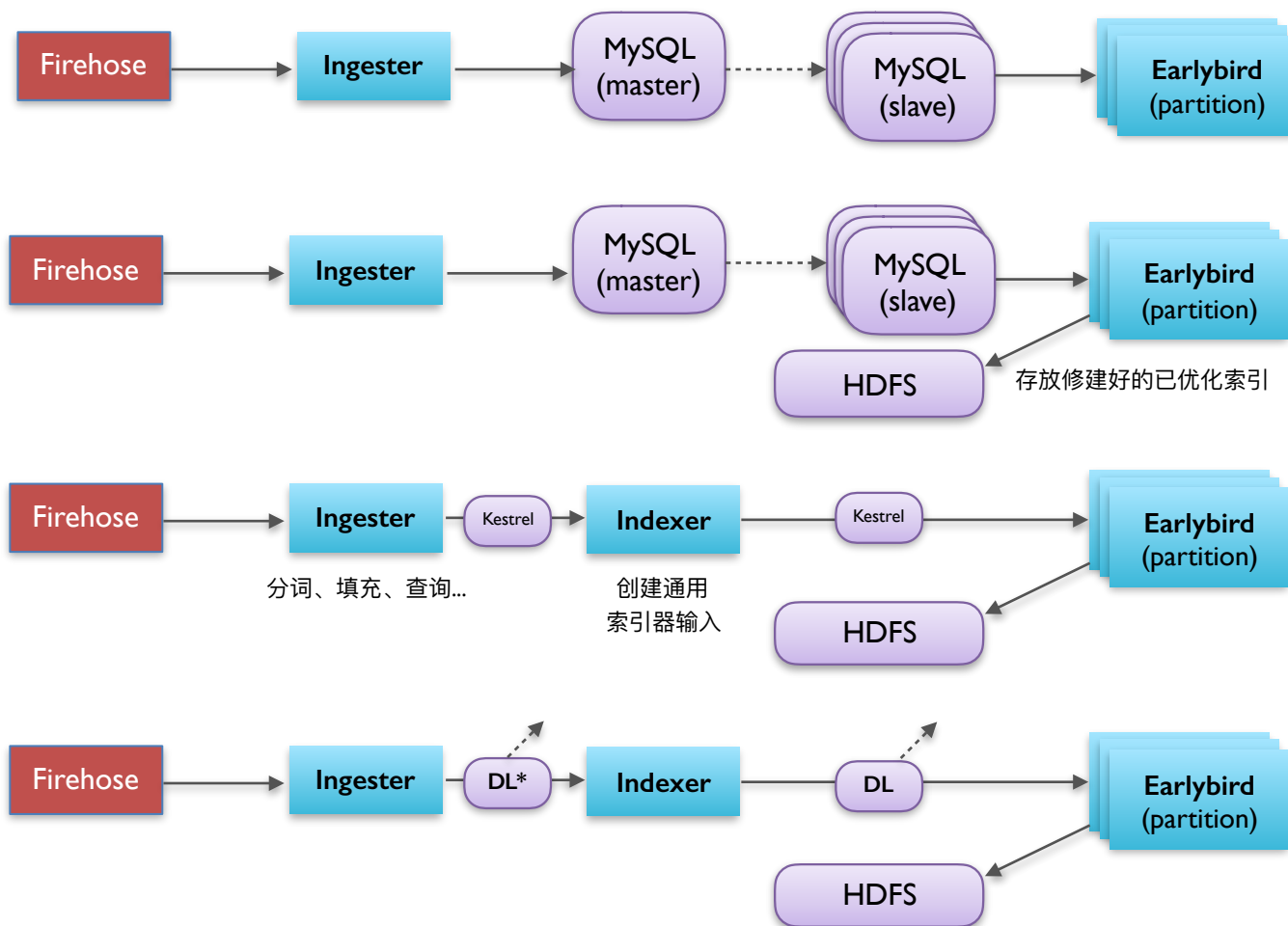


## 视图简化（二）：Superroot

- 提供简单的索引逻辑视图
- 原先分cluster的索引视图对于搜索产品本身的开发非常方便，但是新兴client并不需要着一些信息
- 提供简单的基础搜索功能入口

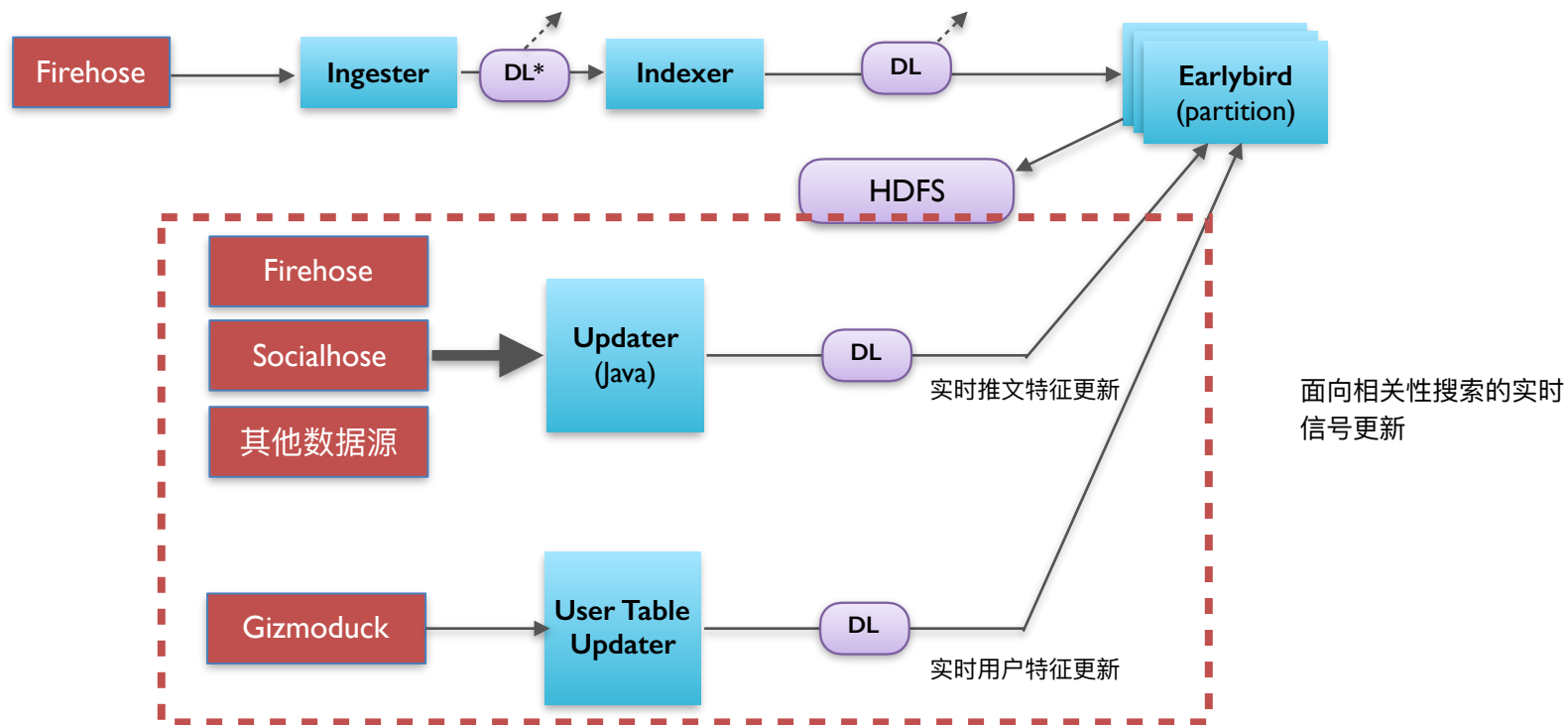


## 索引流水线的演进（一）

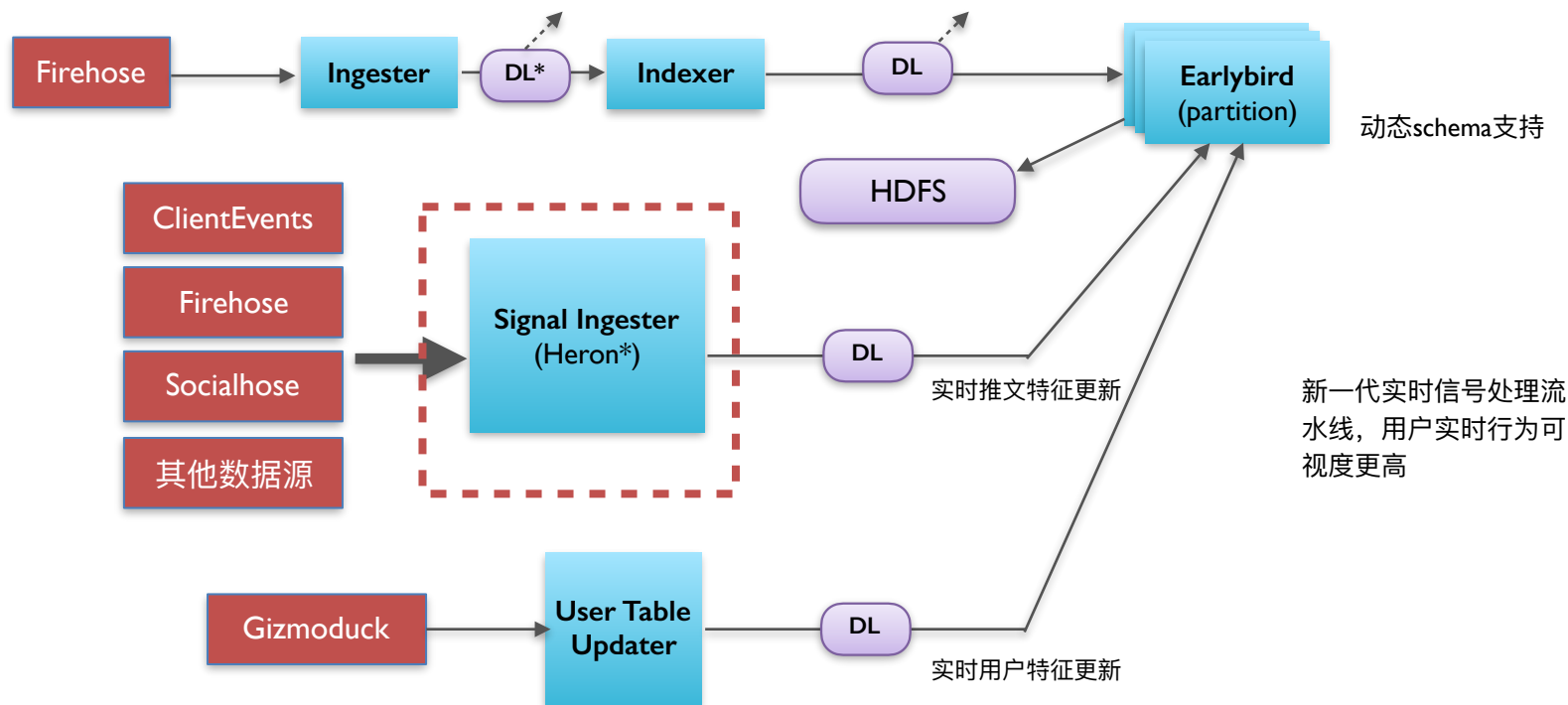


\*DL: DistributedLog, Twitter内部开的基于Apache Bookkeeper和HDFS的队列系统

## 索引流水线的演进（二）



## 索引流水线的演进（三）

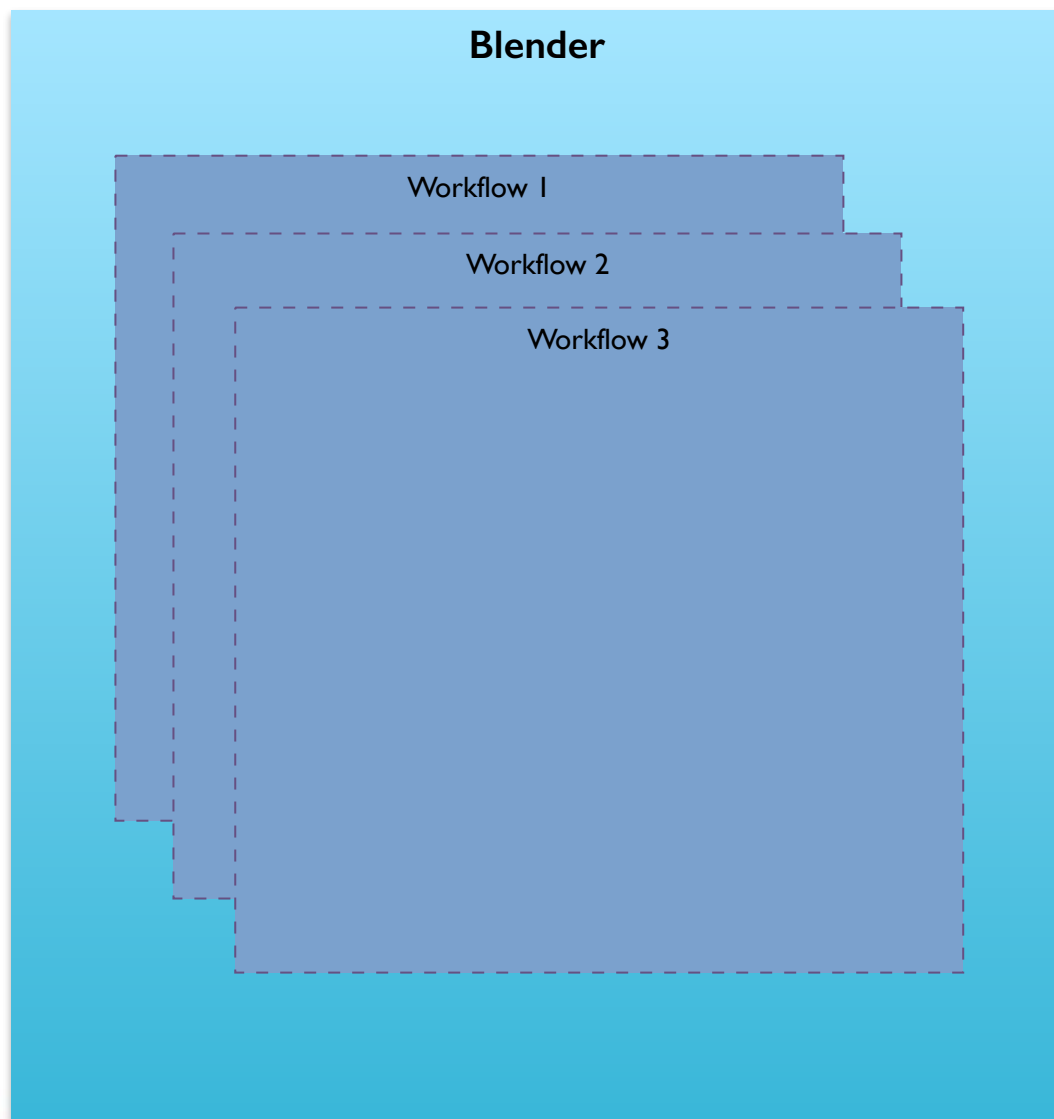


\*Heron 是Twitter开发的新一代和Storm兼容的流计算执行引擎

搜索服务器

# Blender

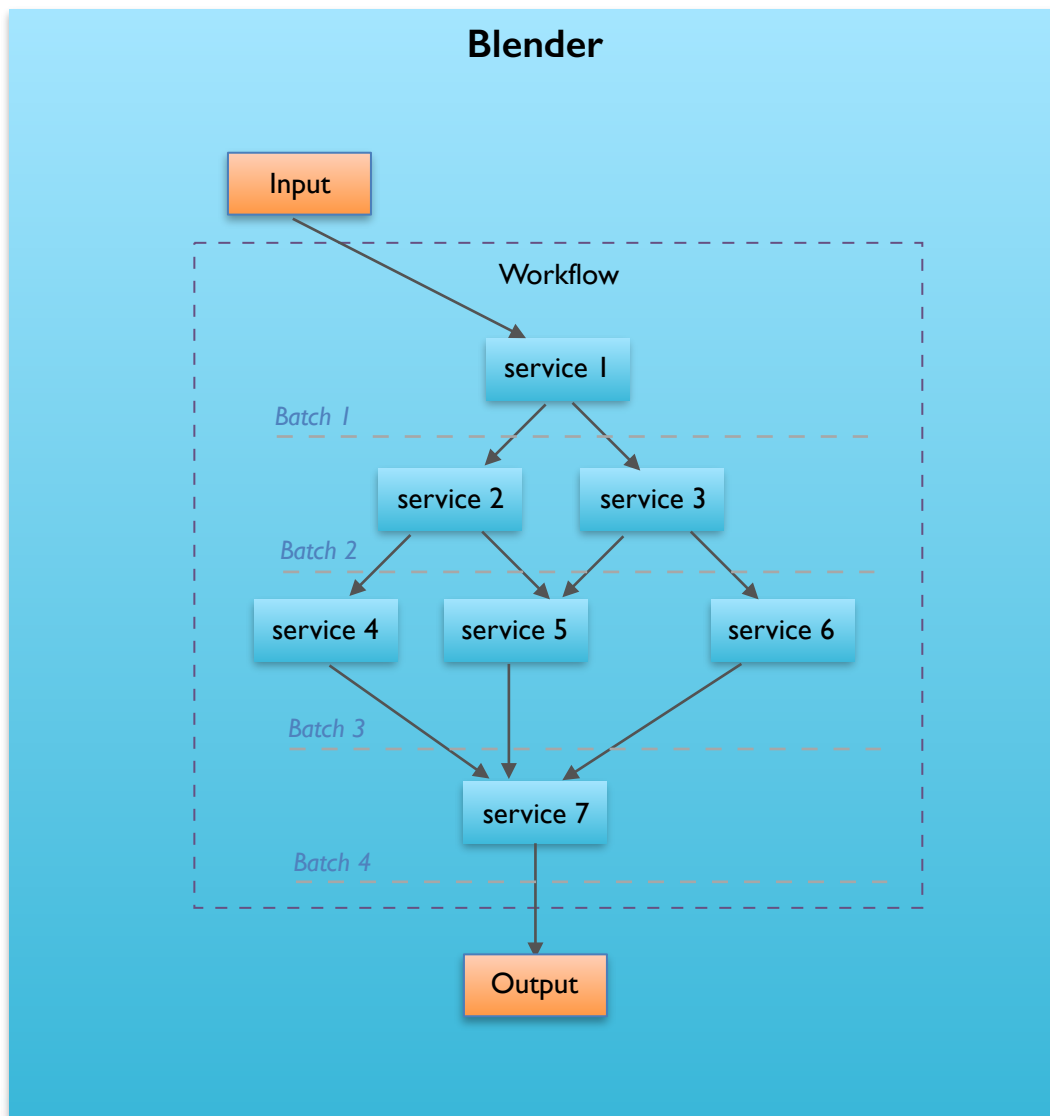
# Workflow



- Blender由多个Workflow组成
  - 同一个Blender可以包含很多完全不同的逻辑步骤
- 每一个Workflow是一个处理输入的特定方式
- 每个Workflow由一个DAG（有向无环图）组成，表示方法经历了若干变化
- Blender的请求中包含Workflow的ID用以选择处理流程

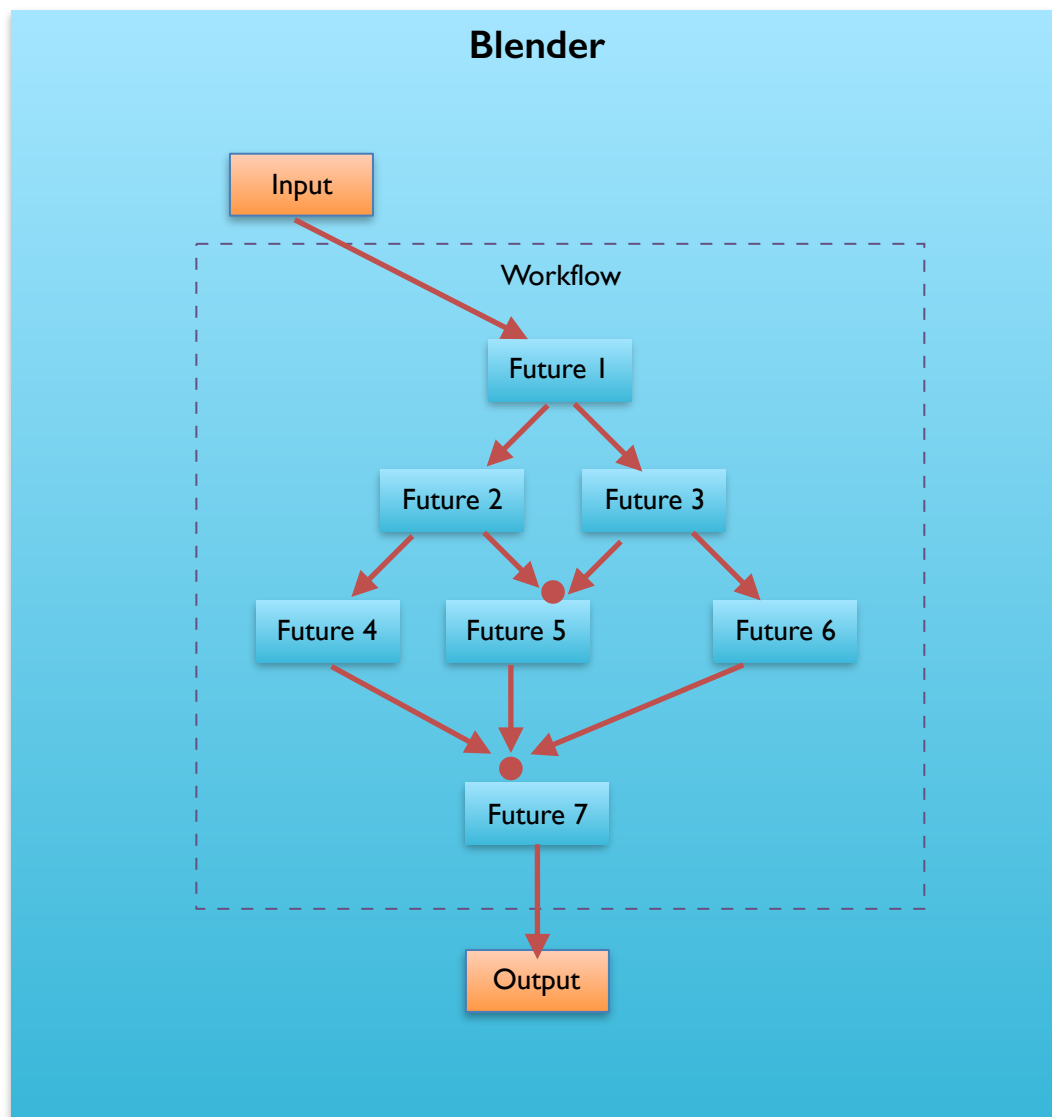


## Blender (一)：基于批次的执行器



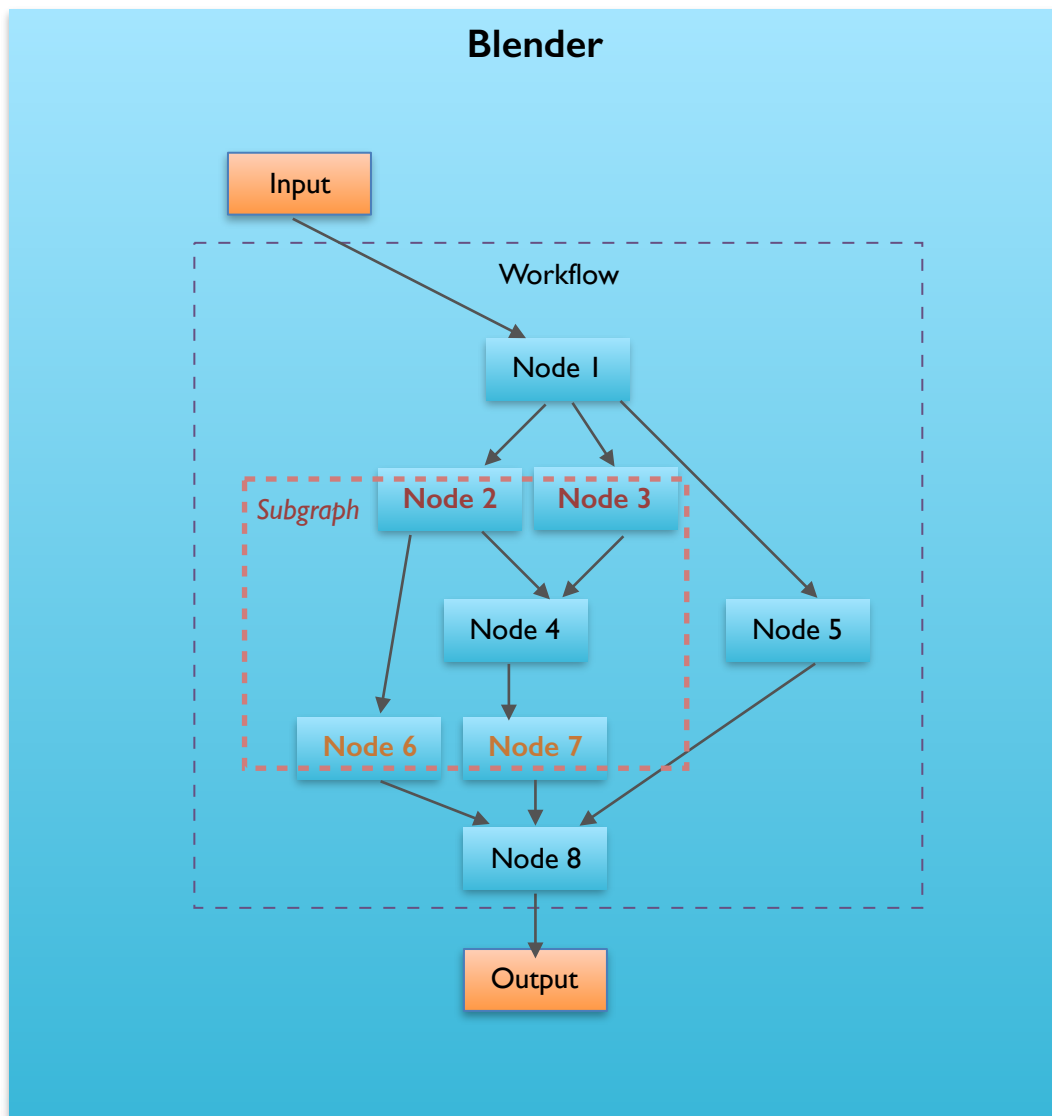
- 依赖关系图（DAG）由代码表示
- 以service调用为节点
- 对DAG拓扑排序后按批次（batch）之行，仅在批次内实现并行化
- 依赖关系清晰
- 实际依赖关系在于数据，而非产生数据的服务
- 服务难以重复调用
- 拓扑排序执行方式效率非最优
- 所有数据全局可见，无法防止任何一步获取它并未依赖的数据

## Blender (二) : 基于Finagle的执行器



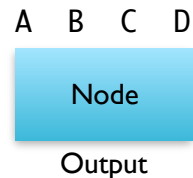
- Finagle是Twitter内部用于规范异步网络编程而出现的软件库
- 提供基于Future链的执行抽象、线程池管理、和失败处理
- 每一个节点是一个Future：对应一份可异步获取的数据
- 每一条边是一个Future操作：调用外部服务，或者进行本地计算
- 所有的操作都是Future上的代数
  - map
  - flatMap
  - collect
  - join
  - ...
- 执行效率高效
- 同步和异步过程得到统一
- 对于大型系统依赖关系变得不明显
- 模块交缠复杂，难以重用
- 难以阅读
- 难以测试

## Blender (三) : Nodes

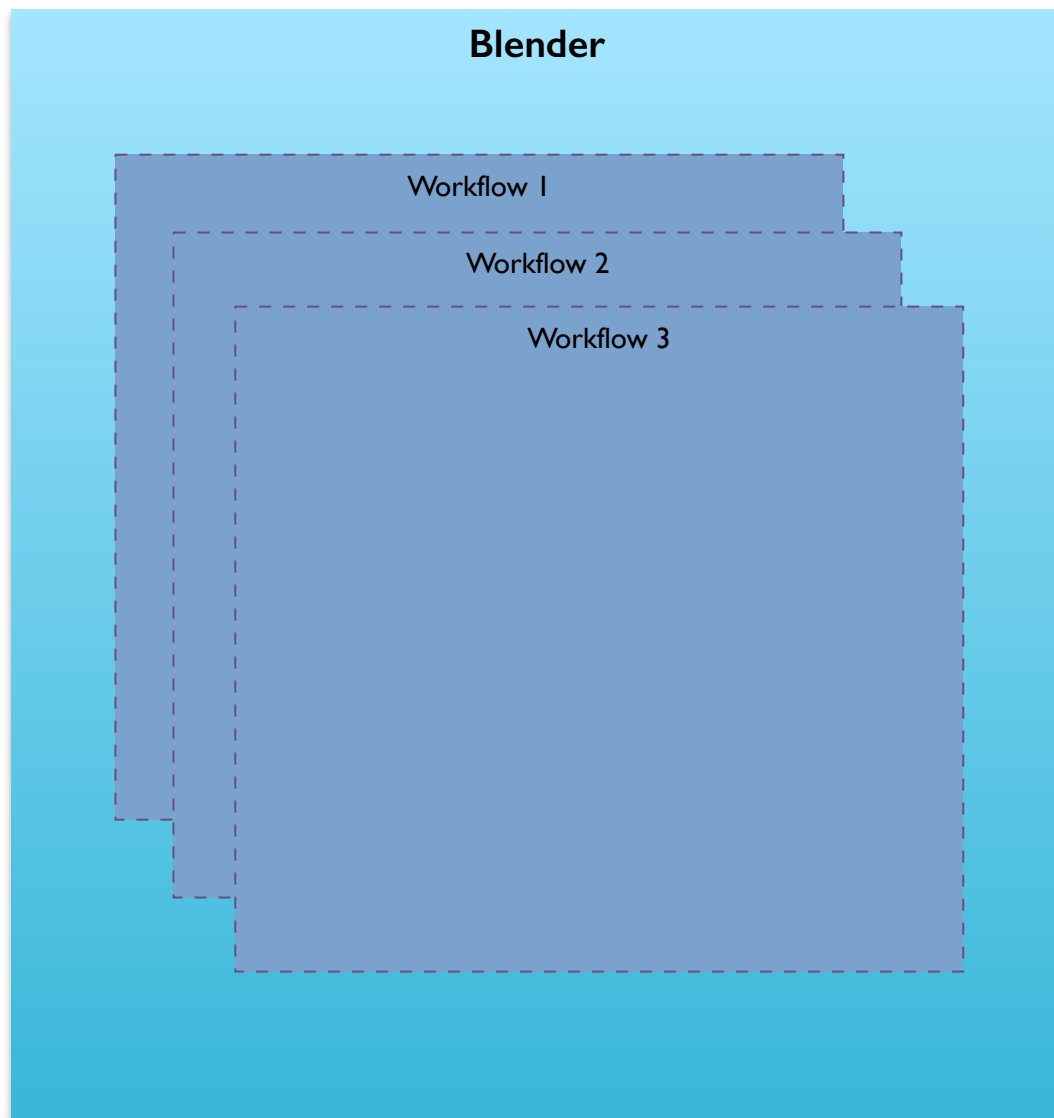


- 进一步抽象和包装Finagle基本Future原语
- **Node**是一个定义了输入依赖和数据类型的异步函数对象，包含多个输入和一个输出
- **Subgraph**是一系列Node的集合，包含位于边界的多个输入和多个输出
- 和第二代代码完全兼容，便于分阶段移植
- 清晰地依赖关系（自动可视化）
- 代码清晰易读
- 便于重用
- 便于测试
- 更强大的自动调试功能
- 和Blender本身解耦，可以用于执行任何依赖关系图
- 执行Overhead略高

一个Node



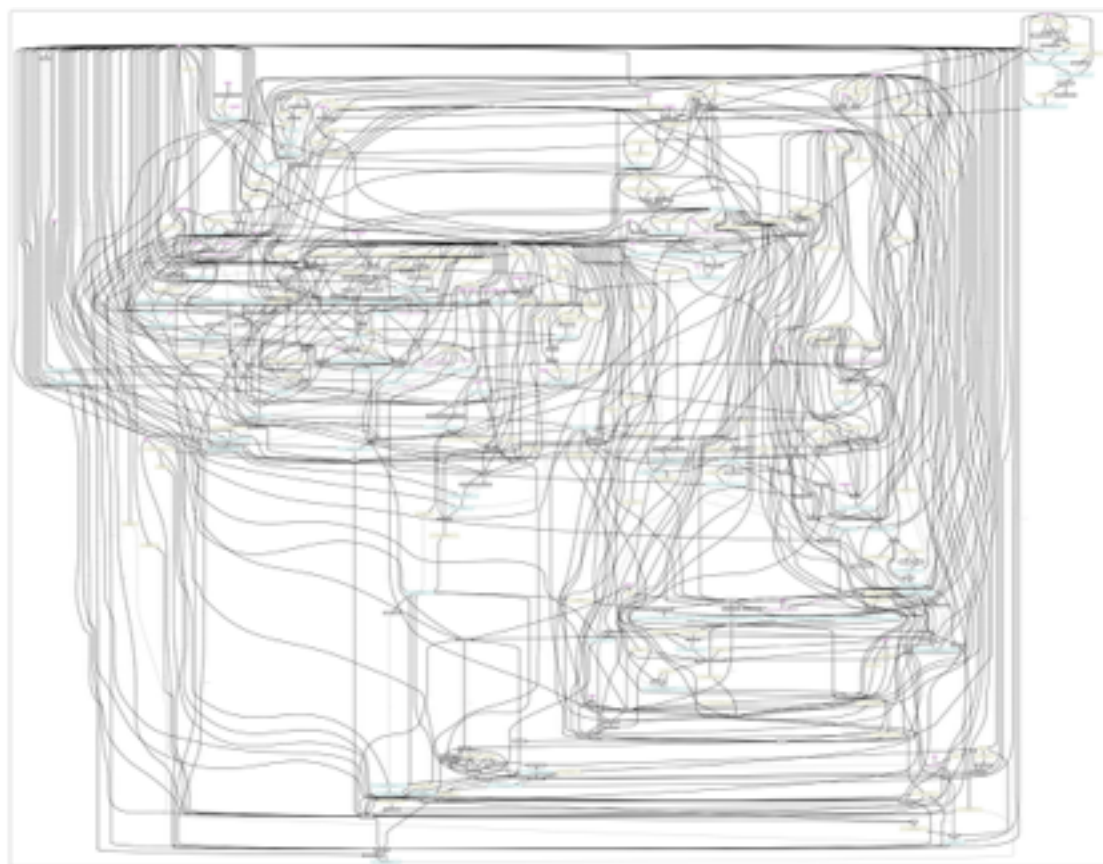
## Workflow的滥觞



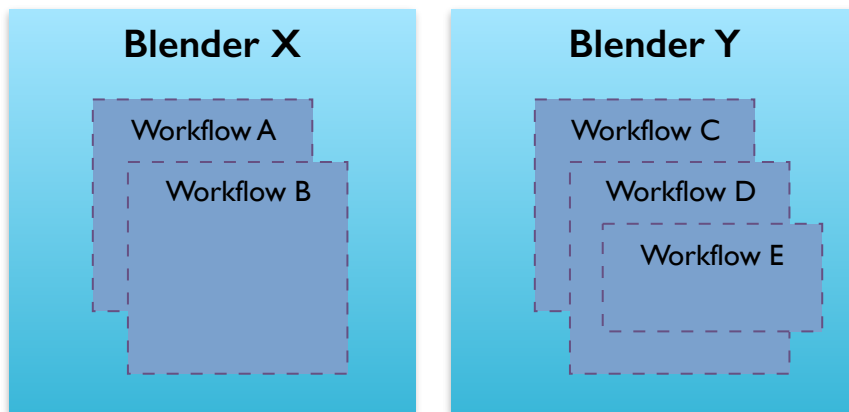
- Workflow的灵活性导致大量被应用
  - 累积了40个不同的workflow
- 因为历史原因，代码重用较差，出现了很多反复书写的类似逻辑
- 解决方法：
  - 参数化Workflow，合并相似过程
  - 组合Workflow，把总是一起调用的Workflow合并为一个
  - 分拆Blender本身\*，独立运营不同的workflow组合

## 最大的Workflow: Adaptive Workflow

- 提供当前Twitter搜索结果页面
- Twitter内最复杂的单个网络请求 (之一?)
- 依赖关系图含有超过1000个节点 (本地和远程)
- 耗时中值230ms左右
- 查询和计算超过15000条结果



## Blender的运营分拆和平台化



- 我们发现Blender各种Workflow用途区别明显，用户期待差异较大
- 搜索部门没有义务运行和维护所有Workflow
- Blender已经成为了便利的开发框架
- 拆分Blender为多个集群
  - 独立开发，Blender核心团队只提供技术指导 and 开发指南
  - 独立部署和运营
  - 独立维护
- 如果只需要运行既有Workflow，只需要配置文件变化即可编译和启动一个新集群

- Blender-adaptive：提供Twitter网站和应用上的搜索功能
- Blender-search：提供面向API和数据服务的搜索
- Blender-typeahead：提供Typeahead（自动补全）索引服务
- ...

# Twitter搜索技术历史

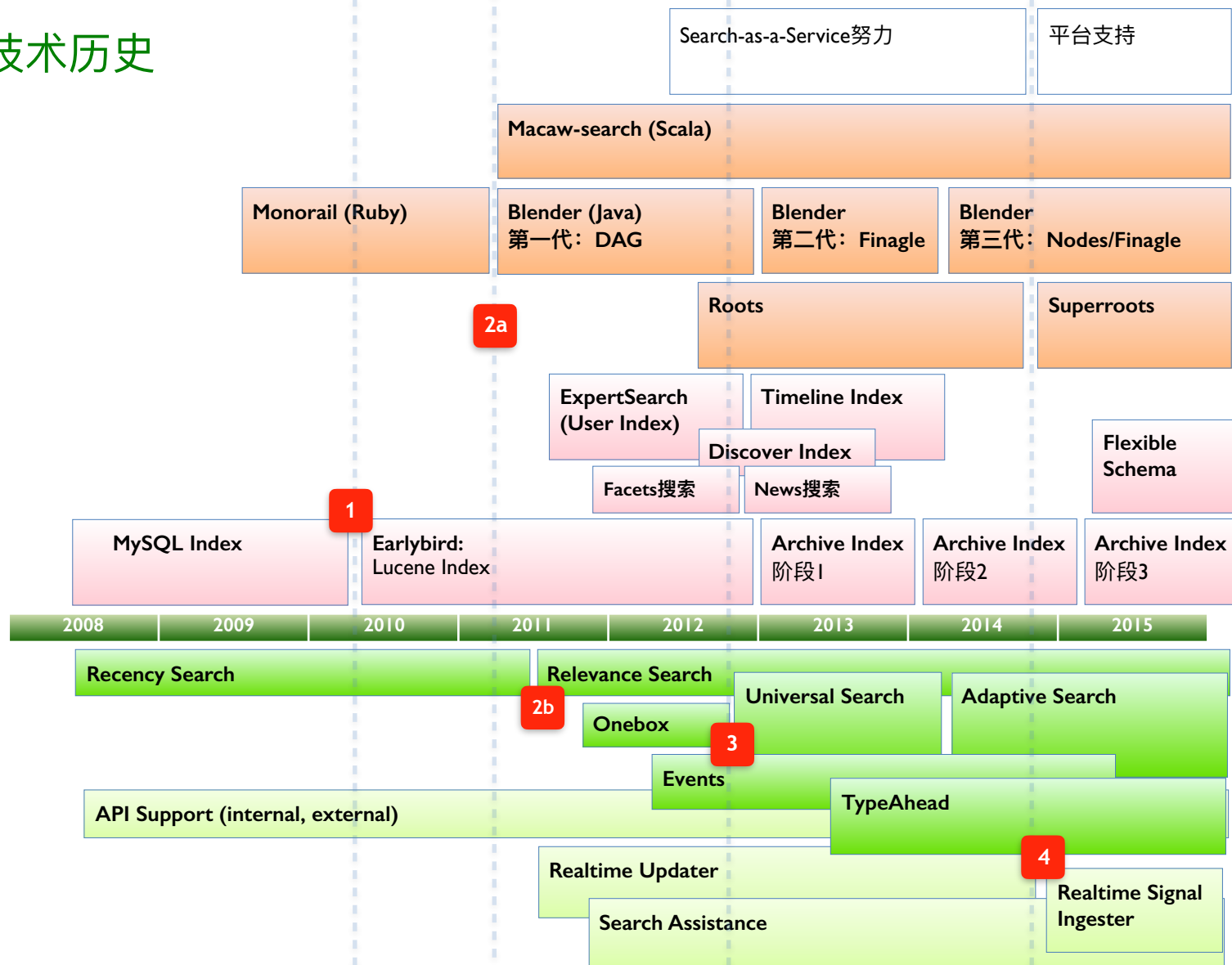
搜索前端

索引服务器

索引流水线

搜索产品

后台支持



## 平台化和技术化

- Earlybird：平台化
  - 我们来开发
  - 我们来运行
  - 多客户需求竞争
  - 你出钱
- Blender：技术化
  - 我们出框架
  - 你自己开发
  - 你自己运行



# 维护质量底线

## 测试

- Unit Test
- Local End-to-end Test
- Smoke Test (请求重现)
- Quality Smoke Test
- Failover Test
- Stress/Redline Test

## 流量应对

- 自动功能退化
- 自动缓存策略
- 请求丢弃/快速失败

## 文档

- 完整的Runbook
- 开发者手册
- 示例代码

## 发布

- 灰度发布
- 试运行 (canary)
- 暗读发布 (dark read)
- 回滚支持
  - 自动生成回滚指南
- 向后兼容发布
- JIRA ticket跟踪全过程
- 用户实验发布 (DDG)

## 监测和预警

- 数据输出
- 图表
- 细化预警规则
- 警报应对指南
- 自动/手动数据中心流量转移

- 得益于标准产品化规范，绝大部分质量保证都无需专门设计

# Q & A

提问时间



# Thanks!

