ArchSummit
全球架构师峰会 2017

主办方 Geekbang 极客邦科技 InfoQ

美团点评用户行为分析系统的构建与优化

孙业锐
美团点评数据平台团队

主办方 Geekbang. 极客邦科技 InfoQ

AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14　北京国际会议中心

扫描关注大会官网

极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新

扫一扫下载极客时间App

SPEAKER
INTRODUCE

孙业锐

美团点评高级技术专家

Apache Kylin PMC

美团点评数据平台查询引擎方向负责人

负责数据生产和查询引擎的改进优化和落地应用

专注于分布式计算，OLAP分析，Adhoc查询等领域

包括但不限于Hive、SparkSQL、Presto、Apache Kylin、Druid等

TABLE OF

CONTENTS 大纲

- 问题分析
- 算法思路
- 工程实现
- 性能优化
- 总结

问题背景

Growth Hacking

转化率

转化率分析（有序漏斗）

路径：首页-搜索-菜品-下单-支付

日期：2017-11-11

时间窗口：1小时

城市：北京

操作系统：iOS

数据探查

UUID	timestamp	page	city	keyword
AAA	100	首页	北京	
AAA	102	搜索页	北京	中餐
AAA	130	菜品页	北京	
BBB	102	首页	北京	
BBB	103	首页	北京	
BBB	140	搜索页	北京	西餐
CCC	101	首页	上海	
CCC	110	菜品页	上海	
CCC	151	搜索页	上海	中餐

直观解法1： Join

几亿条数据的多层Join
代价很高，时间很长

```
select count (distinct t1.id1), count (distinct t2.id2), count (distinct t3.id3) from
(select uuid id1, timestamp ts1 from data where timestamp >= 1510329600 and
timestamp < 1510416000 and page = ‘首页') t1
left join
(select uuid id2, timestamp ts2 from data where timestamp >= 1510329600 and
timestamp < 1510416000 and page = '搜索页' and keyword = '中餐') t2
on t1.id1 = t2.id2 and t1.ts1 < t2.ts2 and t2.ts2 - t1.ts1 < 3600
left join
(select uuid id3, timestamp ts3 from data where timestamp >= 1510329600 and
timestamp < 1510416000 and page = '菜品页') t3
on t1.id1 = t3.id3 and t2.ts2 < t3.ts3 and t1.ts1 < t3.ts3 and t3.ts3 - t1.ts1 < 3600
```

直观解法2：UDAF

几亿个UUID的聚合
没有高效的过滤条件

```
select
funnel(timestamp, 3600, '首页') stage0,
funnel(timestamp, 3600, '首页', '搜索页', keyword = '中餐') stage1,
funnel(timestamp, 3600, '首页', '搜索页', '菜品页') stage2
from data
where timestamp >= 1510329600 and timestamp < 1510416000
group by uuid
```

问题难点

事件有序

序列匹配运算

复杂度超过
普通的集合运算

时间窗口

最大长度约束下的
序列匹配

丰富属性

埋点完全开放

属性基数超百万

维度下钻分析

数据规模

单天日志数百亿条

时间跨度6个月

坏消息

1. 完全随机的漏斗定义
   - 事件组合、时间窗口完全随机
   - 不能实现完全预计算

2. 不同粒度的深入分析
   - 多个层次维度，事件附加属性
   - 需要OLAP下钻和筛选能力

3. 规模与性能的矛盾
   - 在海量数据下实现交互式分析

好消息

1. 支持能力：模式相对确定
   - 核心是集合运算和去重计数
   - 不需要完整的SQL能力

2. 使用场景：查询并发度低
   - 主要是人工探索式分析
   - 可以调度所有资源

3. 数据特点：入库不会修改
   - 可能构建索引

4. 业务特点：指标收敛较快
   - 重点分析转化率偏低的场景
   - 可能快速过滤

问题本质

多维分析和序列匹配下的去重计数

实现目标

实现多维分析和序列匹配下的去重计数、

支持海量数据、交互式响应、

可能利用索引或有限预处理等手段、充分利用资源的

算法和系统

TABLE OF

CONTENTS 大纲

- 问题分析
- 算法思路
- 工程实现
- 性能优化
- 总结

对应策略

根据多个维度做筛选

UUID内事件按照时间排序

漏斗每层节点符合条件的UUID计数

实现多维分析和序列匹配下的去重计数、

支持海量数据、交互式响应、

可能利用索引或有限预处理等手段、充分利用资源的

算法和系统

数据探查

UUID的事件没有排序
序列匹配困难

UUID	timestamp	page	city	keyword
AAA	100	首页	北京	
AAA	102	搜索页	北京	中餐
AAA	130	菜品页	北京	
BBB	102	首页	北京	
BBB	103	首页	北京	
BBB	140	搜索页	北京	西餐
CCC	101	首页	上海	
CCC	110	菜品页	上海	
CCC	151	搜索页	上海	中餐

数据整理

需要遍历每个UUID
维度筛选和序列匹配都很慢

UUID	event1	event2	event3	…… event n
AAA	ts = 100 page = 首页 city = 北京	ts = 102 page = 搜索页 city = 北京 keyword = 中餐	ts = 130 page = 菜品页 city = 北京	
BBB	ts = 102 page = 首页 city = 北京	ts = 103 page = 首页 city = 北京	ts = 140 page = 搜索页 city = 北京 keyword = 西餐	
CCC	ts = 101 page = 首页 city = 上海	ts = 110 page = 菜品页 city = 上海	ts = 151 page = 搜索页 city = 上海 keyword = 中餐	

构建索引

维度对应的UUID集合需遍历
序列匹配更困难

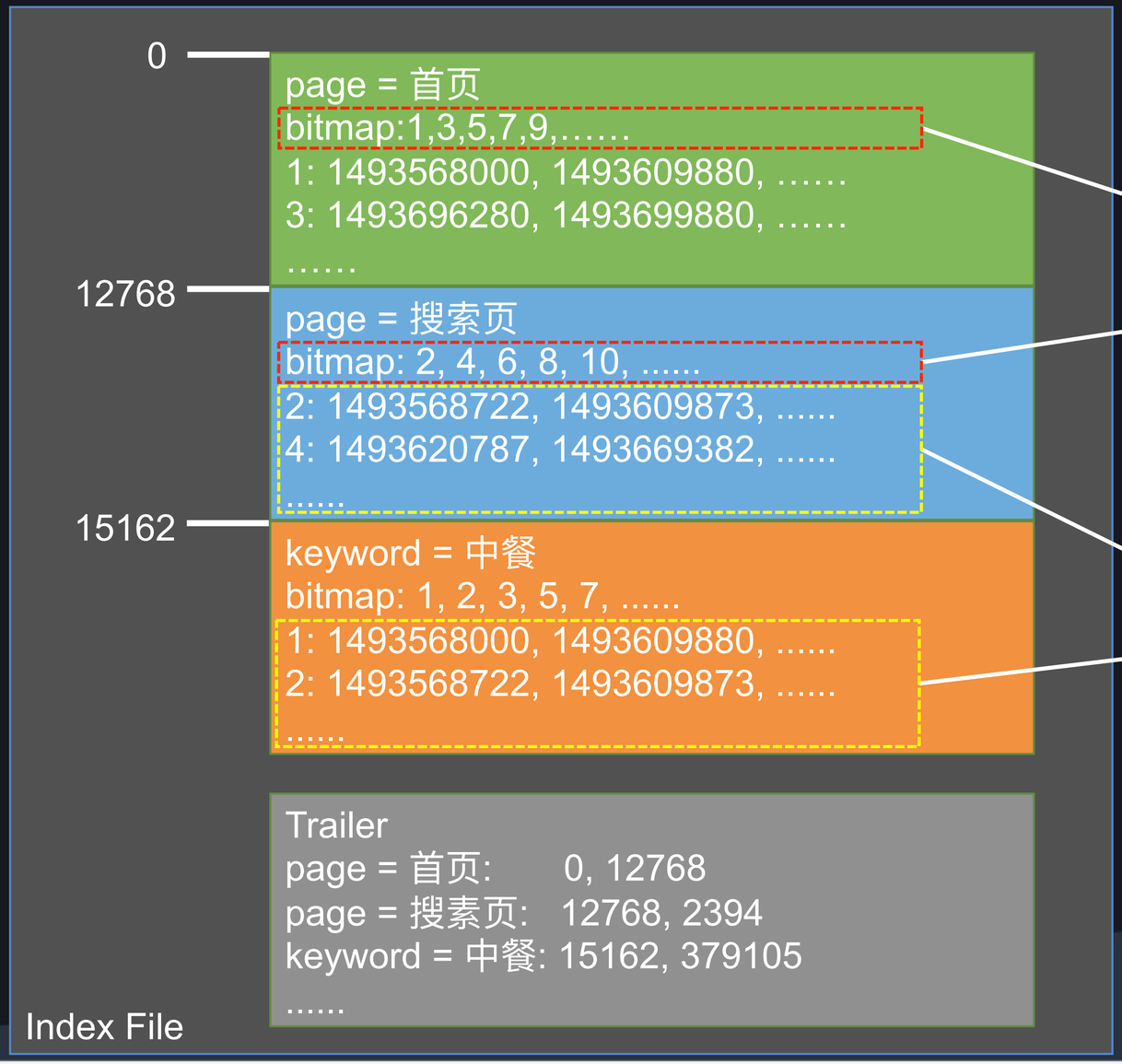
key	value1	value2	value3	…… value n
page = 首页	UUID = AAA ts = 100	UUID = BBB ts = 102	UUID = CCC ts = 101	UUID = BBB ts = 103
page = 搜索页	UUID = AAA ts = 102	UUID = BBB ts = 140	UUID = CCC ts = 151	
page = 菜品页	UUID = CCC ts = 110	UUID = AAA ts = 130		
city = 北京	UUID = AAA ts = 100	UUID = AAA ts = 102	UUID = BBB ts = 102	UUID = BBB ts = 103
city = 上海	UUID = CCC ts = 101	UUID = CCC ts = 110	UUID = CCC ts = 151	
keyword = ……				

索引优化

基于UUID集合快速过滤，迅速收敛
UUID对应的时间序列集中读取

key	UUID collection	sequence
page = 首页	**AAA**,BBB,CCC	**AAA(100)**, BBB(102,103), CCC(101)
page = 搜索页	**AAA**,BBB,CCC	**AAA(102)**, BBB(140), CCC(151)
page = 菜品页	**AAA**,CCC	**AAA(130)**,CCC(110)
city = 北京	**AAA**,BBB	**AAA(100,102,130)**, BBB(102,103,140)
city = 上海	CCC	CCC(101,110,151)
keyword = ……		

索引设计

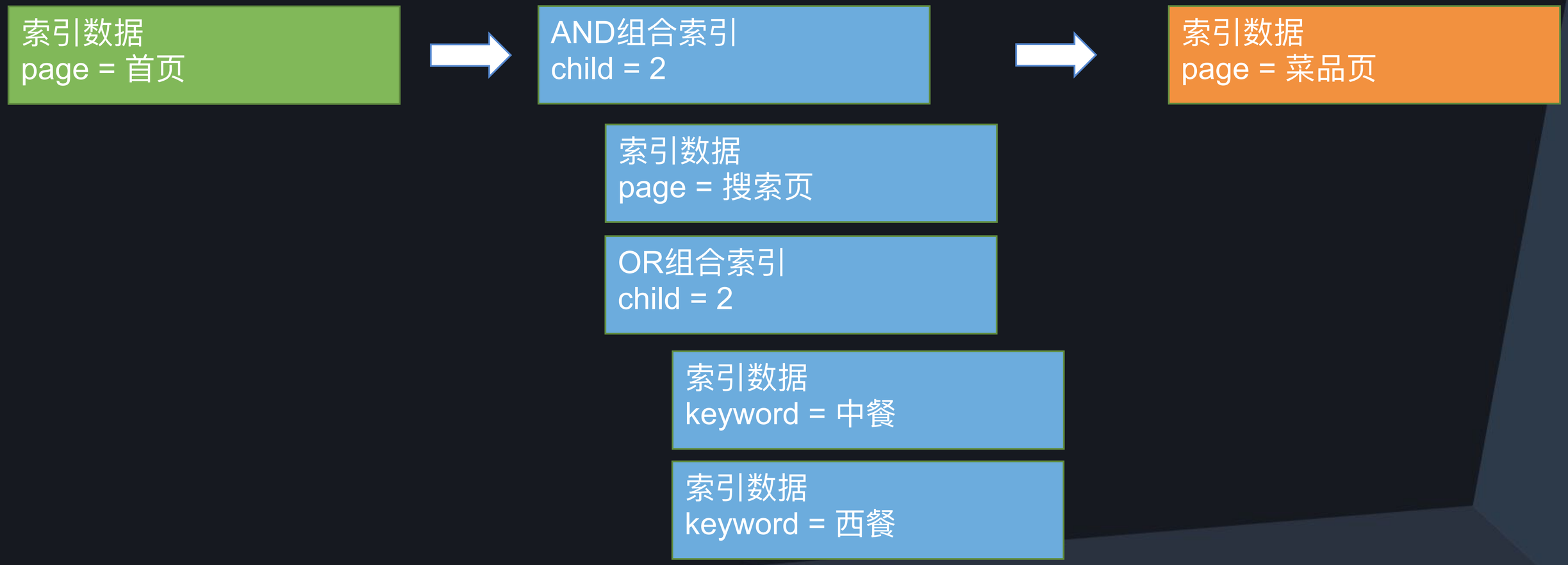


UUID Collection使用Bitmap存储
用于快速过滤数据

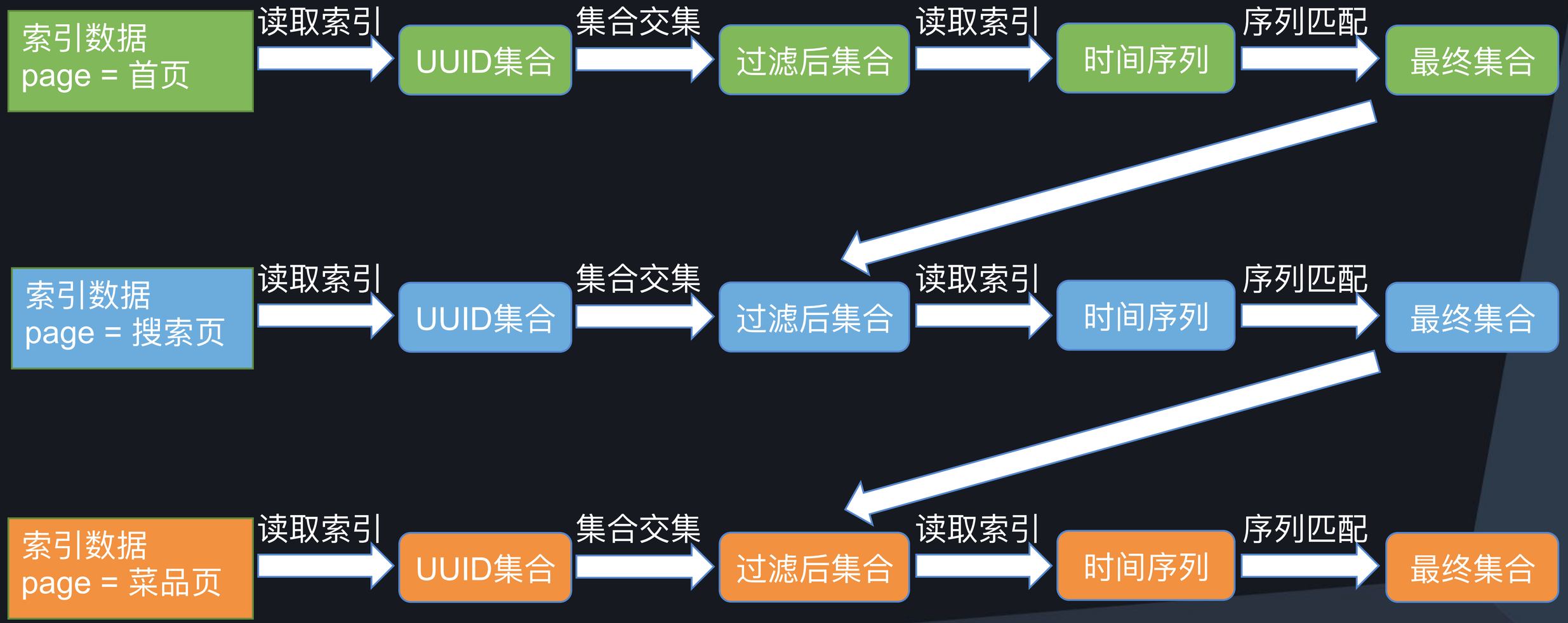
每个UUID的sequence编码存储
用于序列和时间窗口匹配

组合索引

{"expr": "page = 首页"},
{"expr": "page = 搜索页 and keyword in (中餐, 西餐)”},
{"expr": "page = 菜品页"}

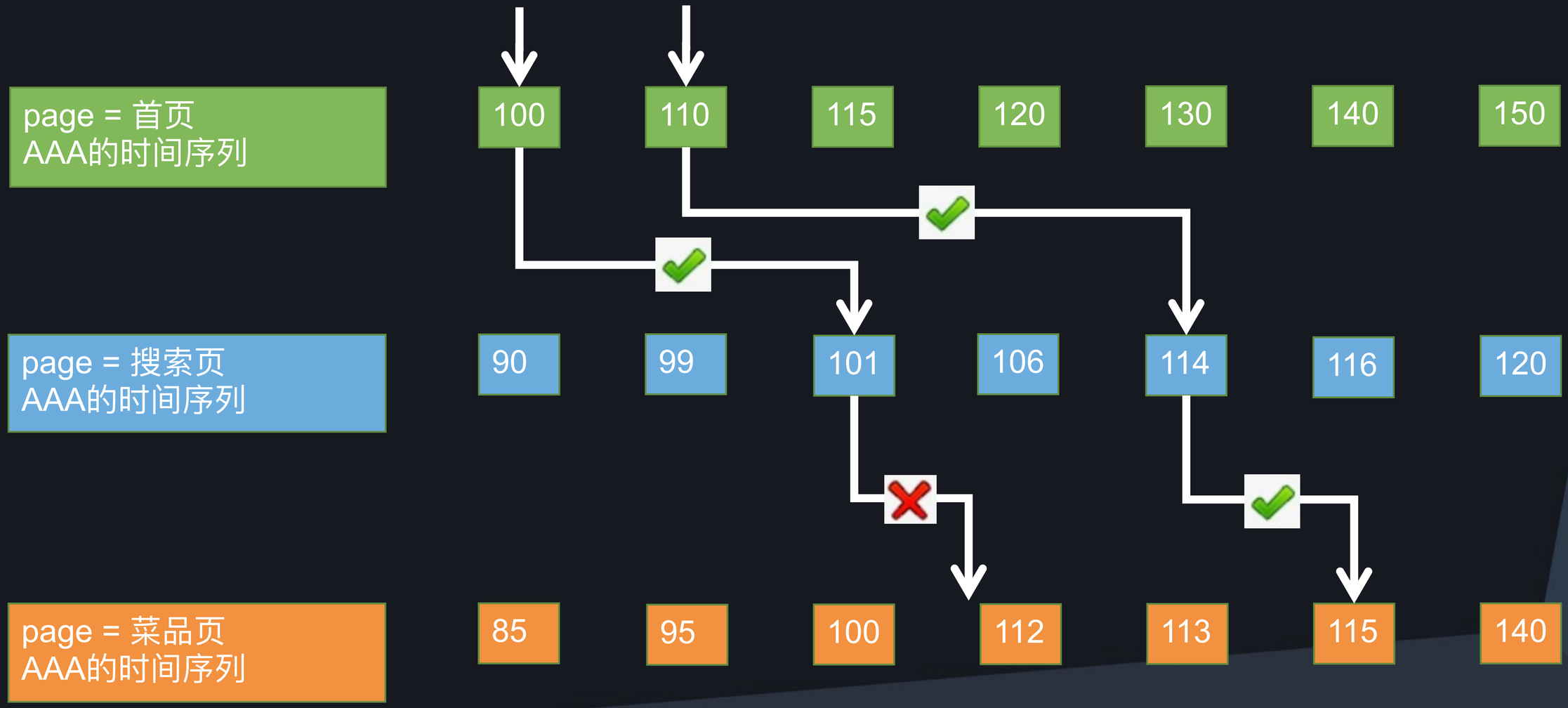


查询过程


索引数据
page = 首页
读取索引
UUID集合
集合交集
过滤后集合
读取索引
时间序列
序列匹配
最终集合
索引数据
page = 搜索页
读取索引
UUID集合
集合交集
过滤后集合
读取索引
时间序列
序列匹配
最终集合
索引数据
page = 菜品页
读取索引
UUID集合
集合交集
过滤后集合
读取索引
时间序列
序列匹配
最终集合

序列匹配

“startTimestamp”: 100, “endTimestamp”: 130, “maxWindow”: 10



核心思路

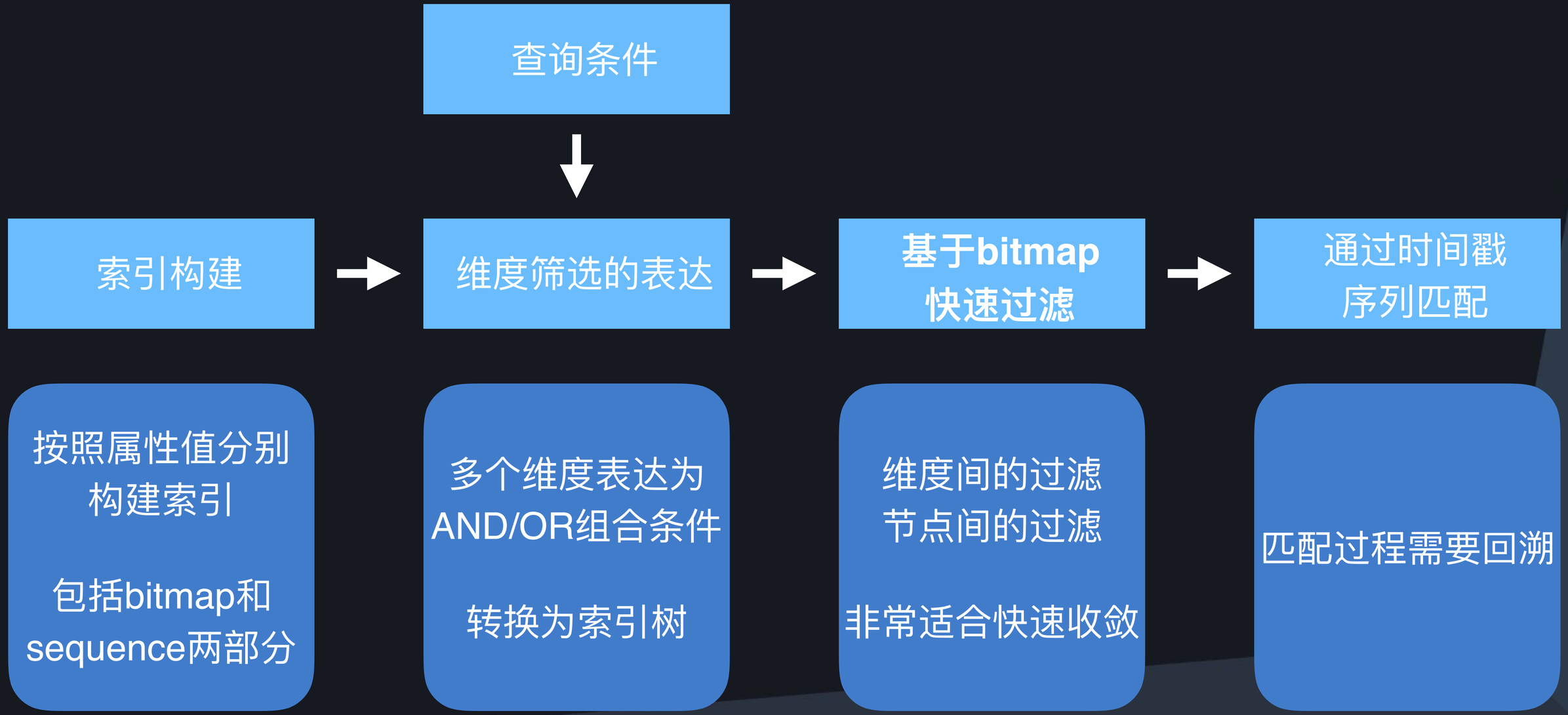

查询条件
索引构建
维度筛选的表达
基于bitmap
快速过滤
通过时间戳
序列匹配
按照属性值分别
构建索引
包括bitmap和
sequence两部分
多个维度表达为
AND/OR组合条件
转换为索引树
维度间的过滤
节点间的过滤
非常适合快速收敛
匹配过程需要回溯

TABLE OF

CONTENTS 大纲

- 问题分析
- 算法思路
- 工程实现
- 性能优化
- 总结

具体需求

- 分布式
- REST服务
- 计算框架
- 文件系统

架构权衡

简单

简单易用

快速落地

成熟

成熟稳定

应用广泛

生态活跃

可控

掌控能力

深度定制

可调

工程优化

持续迭代

实际选择

Spring

应用广泛
文档丰富
简单易用

Netty/Jetty

Spark

分布式调度框架
掌控力强
可高度定制
专注逻辑

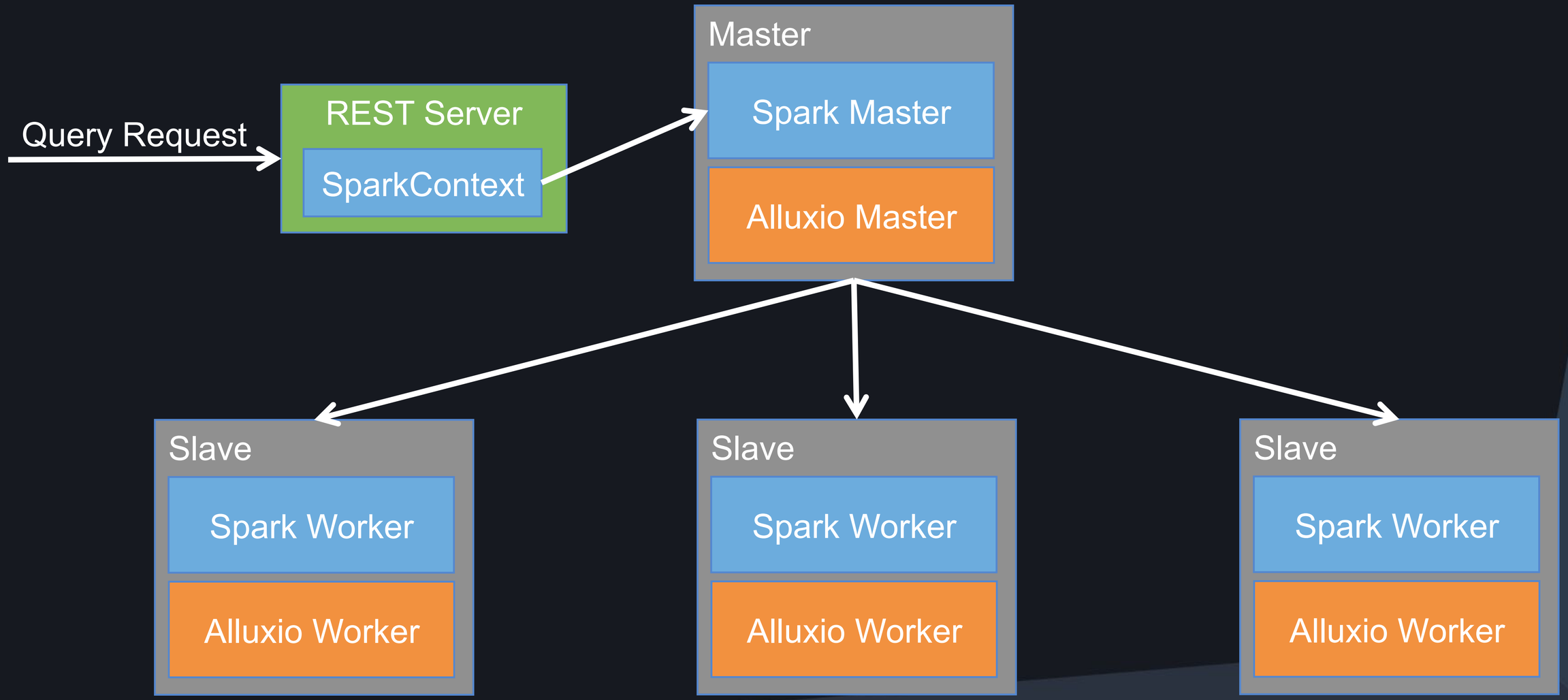
MapReduce

Alluxio

部署简单
相对轻量级
异构存储
性能优化空间

HDFS/HBase

整体架构


Query Request
REST Server
SparkContext
Master
Spark Master
Alluxio Master
Slave
Spark Worker
Alluxio Worker
Slave
Spark Worker
Alluxio Worker
Slave
Spark Worker
Alluxio Worker

工程要点

查询条件 → 属性表达树的组合 → 基于Bitmap快速过滤 → 通过时间戳序列匹配

采用Json格式REST服务接收请求

Antlr解析条件生成表达树

使用RoaringBitmap易用，快速

采用差值存储平均长度由8字节降到1.5字节

TABLE OF

CONTENTS 大纲

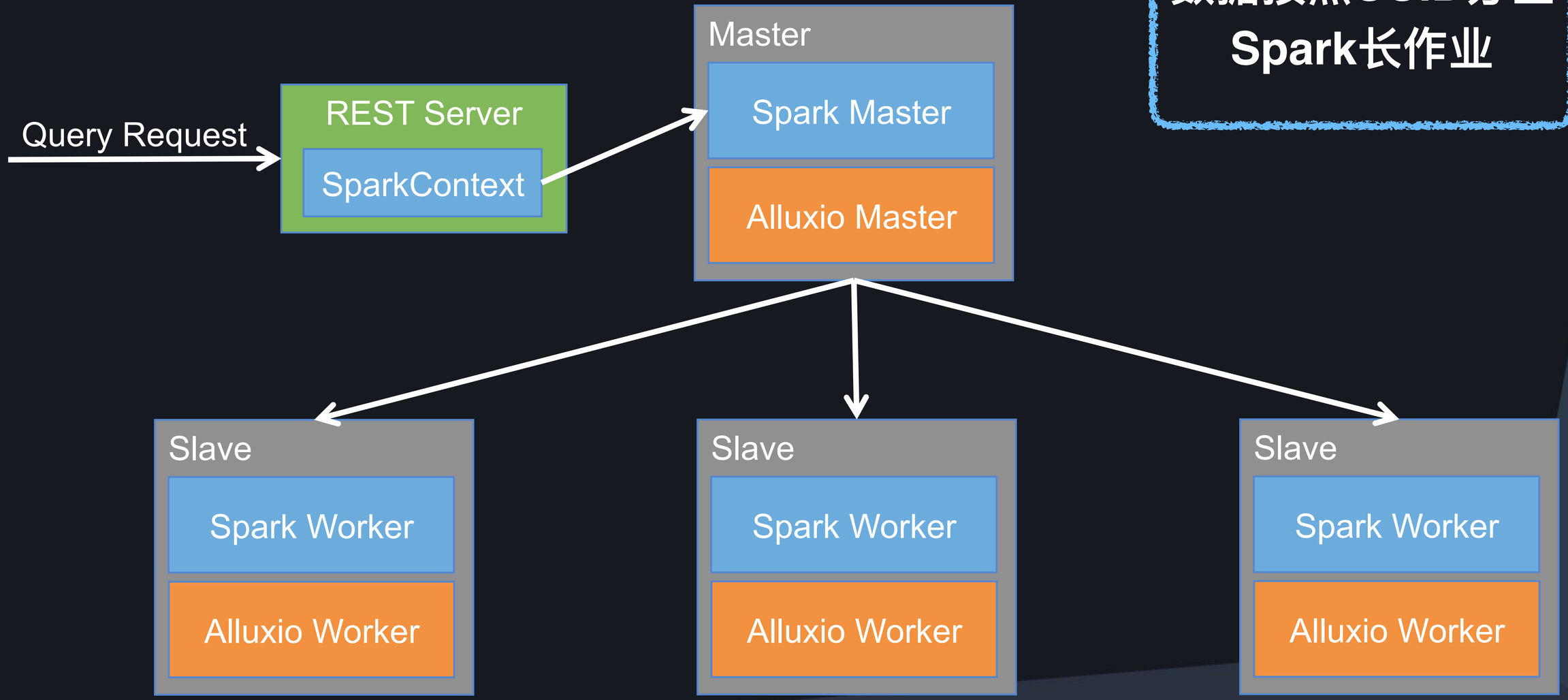
- 问题分析
- 算法思路
- 工程实现
- 性能优化
- 总结

核心关注

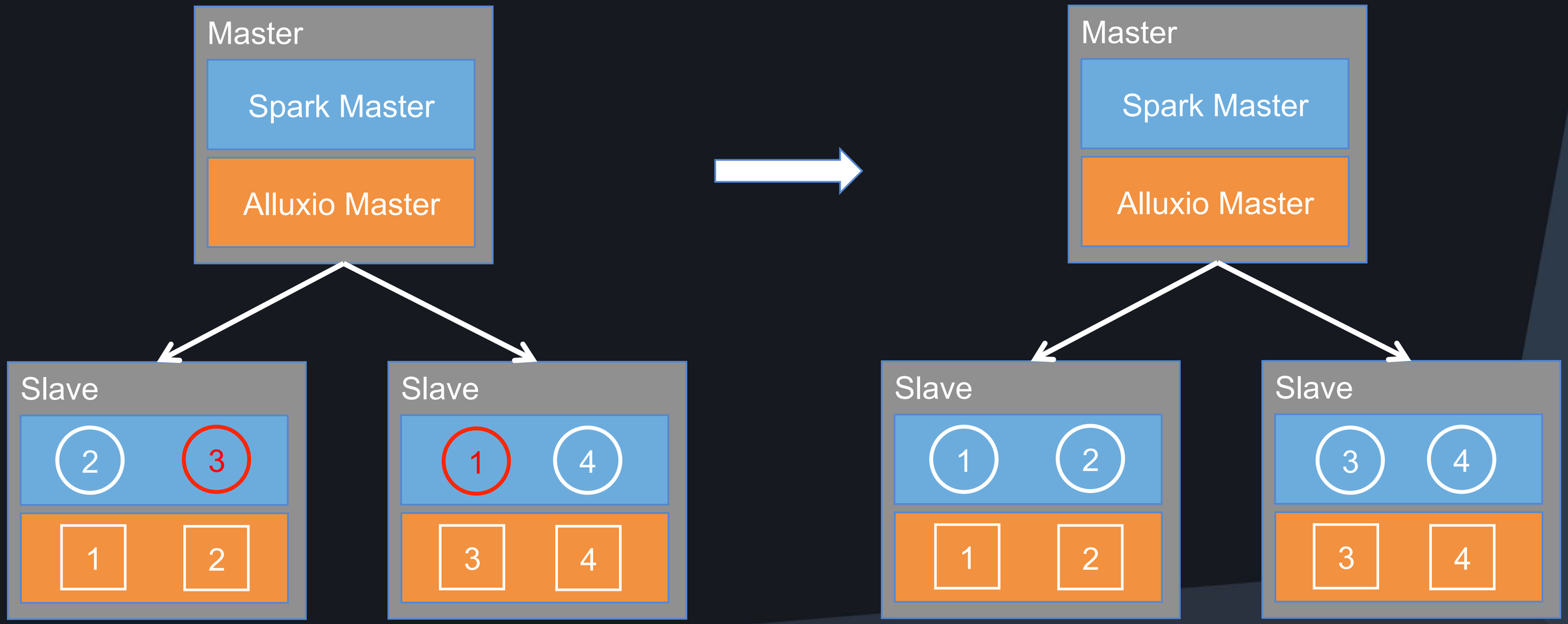
眼镜	尺子	资源利用率	JVM
有效Profiling手段 JStack/JStat JMap/JHat MAT/JMC	可量化的指标 时延/吞吐	CPU 内存 磁盘 网络	内存 GC 方法热点

基本优化：几分钟

数据按照UUID分区
Spark长作业

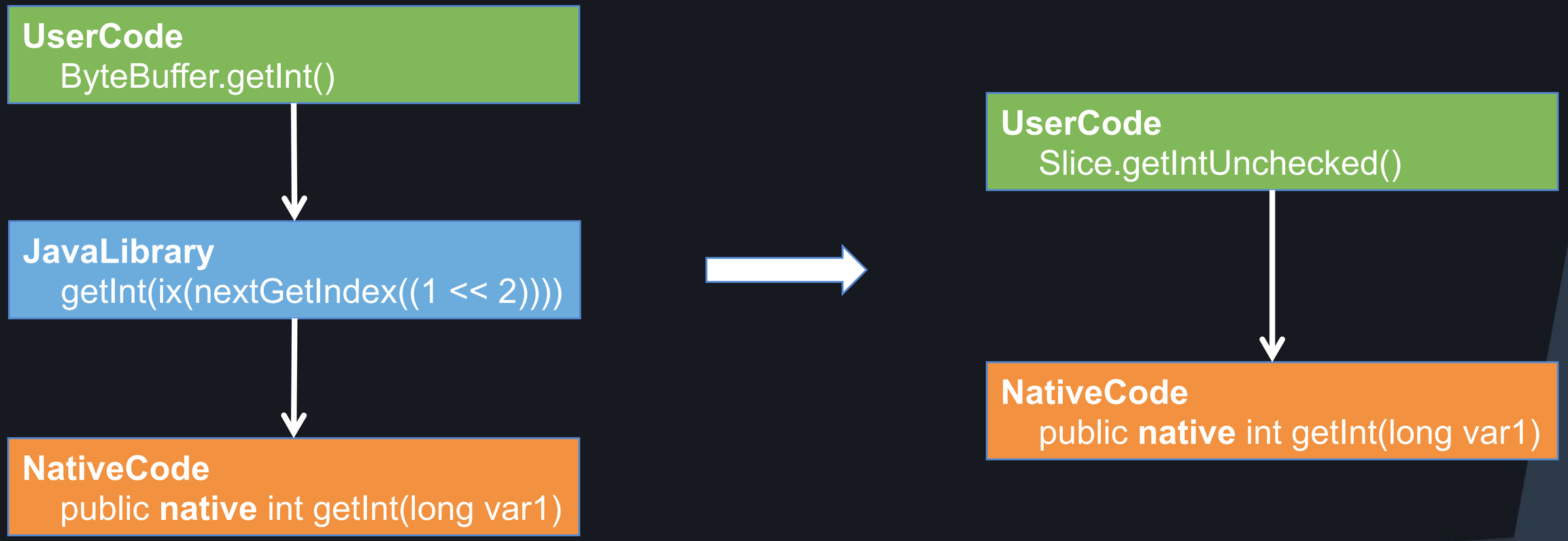


本地化调度：分钟内


Master
Spark Master
Alluxio Master
Slave
2
3
1
2
Slave
1
4
3
4
Master
Spark Master
Alluxio Master
Slave
1
2
1
2
Slave
3
4
3
4

内存映射：10秒内

Unsafe调用：5秒内



优化历程

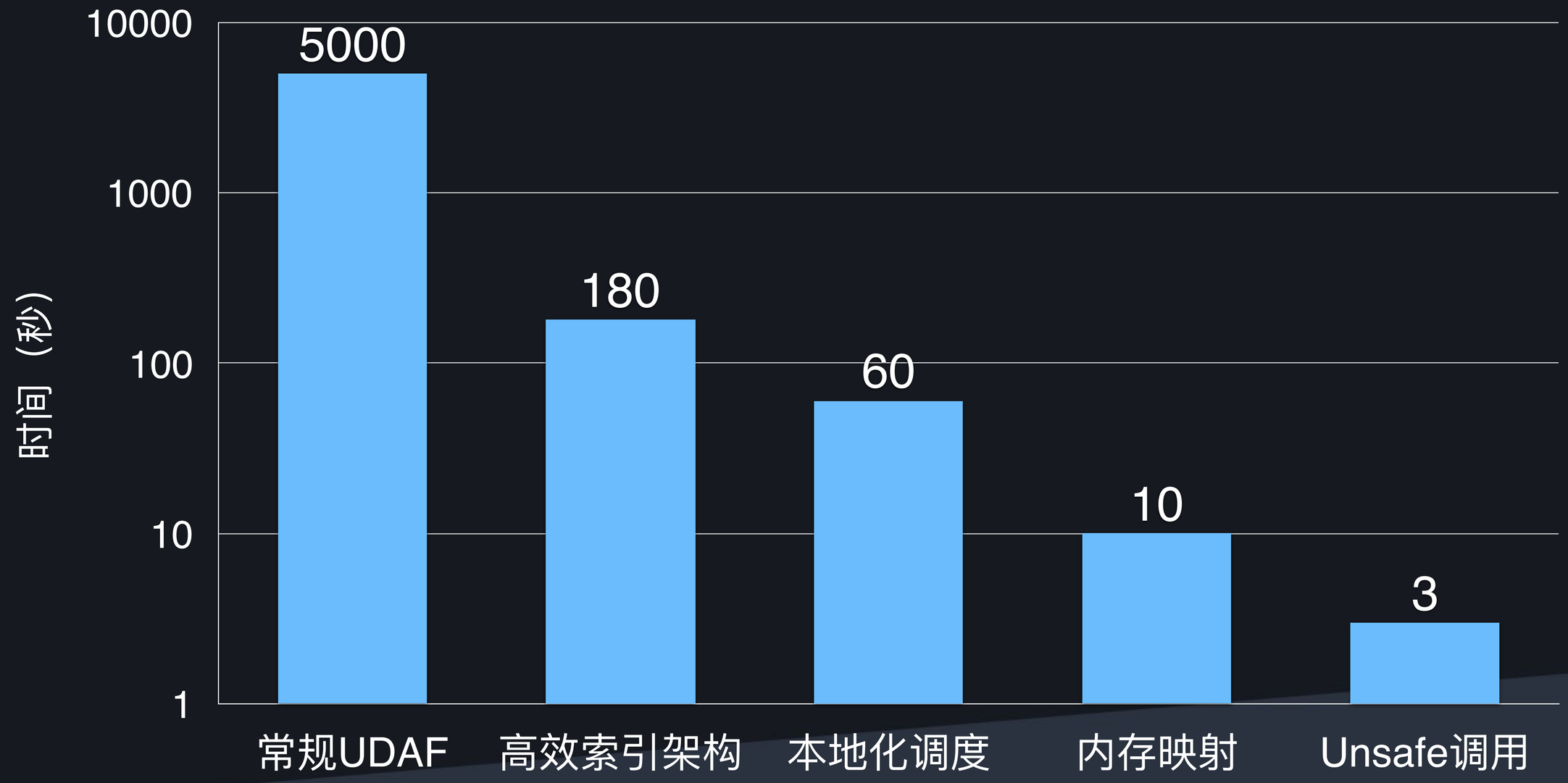

10000
1000
100
10
1
时间（秒）
5000
180
60
10
3
常规UDAF
高效索引架构
本地化调度
内存映射
Unsafe调用

TABLE OF

CONTENTS 大纲

- 问题分析
- 算法思路
- 工程实现
- 性能优化
- 总结

发展现状

数百亿
事件

亿级UV

超百万
属性埋点

数百次
查询

**TP95
小于5秒**

方法总结

项目阶段	方法要点	实际应用
需求分析	理解问题本质 基于本质确定目标	有序漏斗的本质理解 在此基础上确定实现目标
算法设计	正反分析，确定边界 问题拆解与转化，方案借鉴	坏消息和好消息 直观解法入手，逐步优化思路
工程实现	选型权衡的原则和方法	简单、成熟、可控、可调
优化迭代	眼镜和尺子 持续抓瓶颈	Profiling手段和可量化的指标 常见的可能瓶颈和对应优化方法

未来规划

- 代码开源，社区共建

- 功能与性能的迭代
  - 留存统计和路径分析
  - 更高的执行效率
  - 更紧凑的存储格式
  - 更合理的系统架构

加入我们

- 我们有

  近万台机器，数百PB数据

  数十业务线，数百分析师

  业界突破：HDFS异地跨机房，YARN调度百倍性能提升，Spark核心性能优化

- 我们要

  追求极致，持续深耕的领域专家

  充满激情，挑战自我的技术新兵

- 联系方式：sunyerui@{meituan.com, gmail.com, apache.org}

THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

ArchSummit
全球架构师峰会 2017