

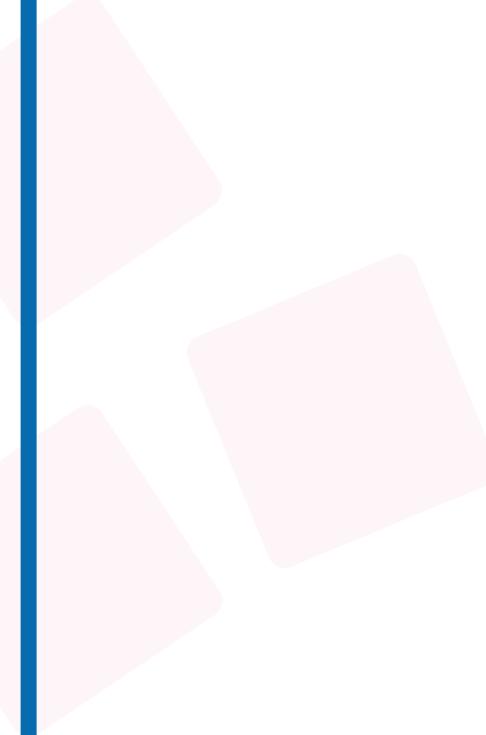
Kylin在用户行为分析场景的应用

俞霄翔 Kylin Contributor

2019.11

议程





技术原理

从Bitmap到留存分析

站点分析

现在有一个 app 的用户访问记录表 `access_log`, 它包含三个字段: DT (访问日期), User ID (用户标示) 和 Page (访问页)。

在网站/app 使用统计中, PV/UV 是最常用的指标, 其中 UV (unique visitor, 独立访问用户) 就是去重后的数字, 即同一个用户的所有访问记录只计入一次。对于网站/app 所有者, PV (page view) 代表的使用量的高低, UV 代表用户的多少, 两个数字都很重要; 只有结合两个数字一起, 才能更加准确地了解网站/app的用户、用量增长情况。

日期	User ID	Page ID
20190101	100	index.html
20190101	102	search.html
20190101	100	detail.html
20190102	100	detail.html
20190102	102	search.html
20190102	100	search.html



站点分析

现在有一个 app 的用户访问记录表 `access_log`, 它包含三个字段: DT (访问日期), User ID (用户标示) 和 Page (访问页)。

在网站/app 使用统计中, PV/UV 是最常用的指标, 其中 UV (unique visitor, 独立访问用户) 就是去重后的数字, 即同一个用户的所有访问记录只计入一次。对于网站/app 所有者, PV (page view) 代表的使用量的高低, UV 代表用户的多少, 两个数字都很重要; 只有结合两个数字一起, 才能更加准确地了解网站/app的用户、用量增长情况。

日期	User ID	Page ID
20190101	100	index.html
20190101	102	search.html
20190101	100	detail.html
20190102	101	detail.html
20190102	102	search.html
20190102	100	search.html



日期	Set[User ID]
20190101	100,102
20190102	100,101,102

留存分析

漏斗分析，又叫转化漏斗，就是将一个特定过程的多个步骤间的转化情况，以漏斗的形式展示出来，通过图形直观地发现流失最严重的环节，从而有针对性地去进行优化。

我们希望回答运营同学这样的疑问：有多少访问了首页的用户，进入到了产品详细页？看了产品详情页的，有多少用户将它加入到了购物车？

日期	UserID	PageId
20190101	100	index.html
20190101	100	search.html
20190101	100	detail.html
20190102	101	index.html
20190102	101	search.html
20190102	102	index.html



留存分析

漏斗分析，又叫转化漏斗，就是将一个特定过程的多个步骤间的转化情况，以漏斗的形式展示出来，通过图形直观地发现流失最严重的环节，从而有针对性地去进行优化。

我们希望回答运营同学这样的疑问：有多少访问了首页的用户，进入到了产品详细页？看了产品详情页的，有多少用户将它加入到了购物车？

日期	UserID	PageId
20190101	100	index.html
20190101	100	search.html
20190101	100	detail.html
20190102	101	index.html
20190102	101	search.html
20190102	102	index.html



页面	Set[UserID]
index.html	100,101,102
search.html	100,101
detail.html	100

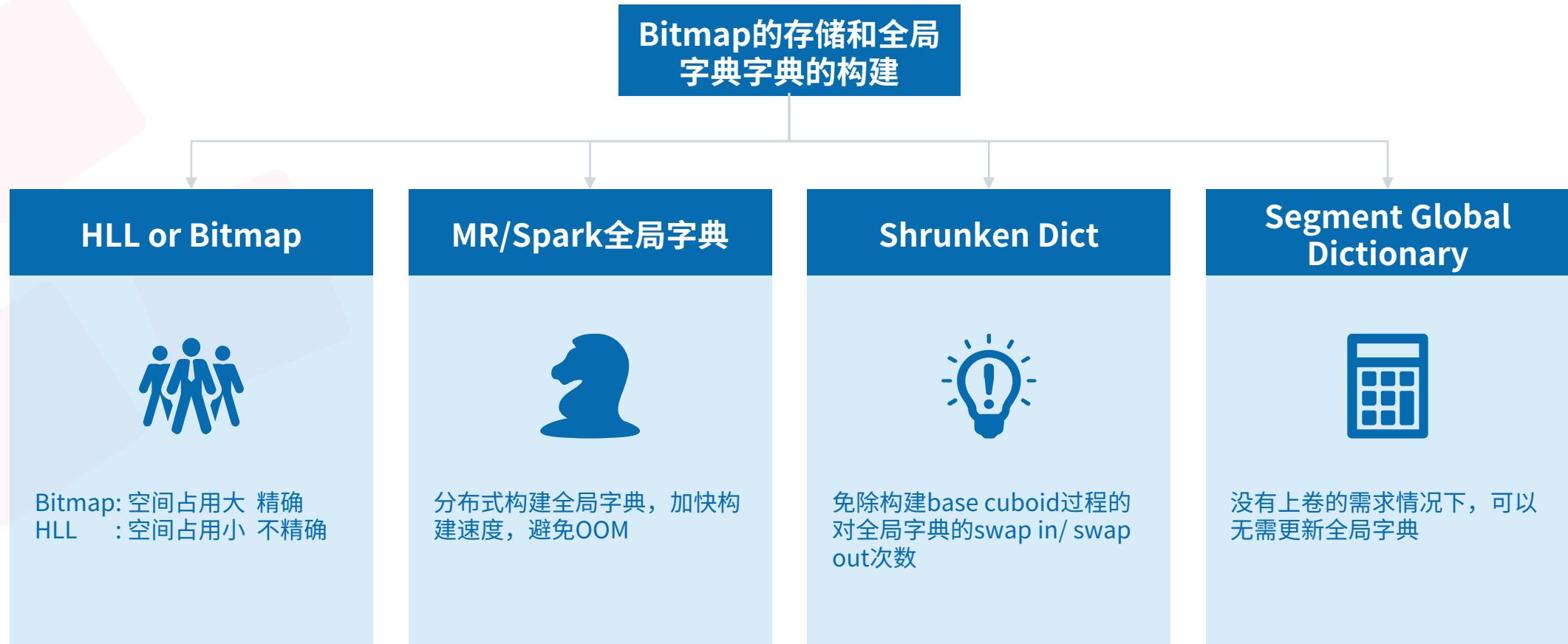


页面	Set[UserID]	UV
index.html	100,101,102	3
index.html & search.html	100,101	2
index.html & search.html & detail.html	100	1

构建流程



挑战



Bitmap的多种用途

Bitmap 本质上是一种集合，可以执行集合之间的操作，包括交并补差等操作。

通过这些集合操作，可以高效快捷地满足很多业务场景。

Kylin目前通过 `intersect_count` 来支持支持集合的交操作；未来可能支持 `intersect_value` 或者更多集合操作等操作。





实际操作



简单清晰

计算代价低

留存分析

计算第一天访问的用户中，有多少在第二天、
第三天继续访问了 app。如果使用 HiveQL
或 Spark SQL 来计算第一天和第二天的留存

用户数，写法大致如下：

```
SELECT count(distinct first_day.USER_ID) FROM
  (select distinct USER_ID as USER_ID from access_log where DT = '20190101') as first_day
  INNER JOIN
  (select distinct USER_ID as USER_ID from access_log where DT = '20190102') as second_day
    ON first_day.USER_ID = second_day.USER_ID
```

使用 Kylin 来实现，SQL 的实现如下：

```
SELECT intersect_count(user_id, dt, array['20190101', '20190102'])
FROM access_log
WHERE dt IN ('20190101', '20190102')
```

滑动留存分析

现实中有时候需要在多个维度上同时进行滑动分析，例如运营可能会问：第一天访问“商品明细页”的用户，有多少在第二天访问了“付款页”？

虽然 `intersect_count` 交集函数只接受一个维度值的变化，但我们可以巧妙利用 `where` 做其它维度的筛选，最后的结果交给 SQL 执行器来计算。

日期	UserID	PageId
20190101	100	index.html
20190101	102	search.html
20190101	100	detail.html
20190102	100	detail.html
20190102	102	search.html
20190102	100	search.html

```
select
    intersect_count(user_id, dt, array['20190101']),
    intersect_count(user_id, dt, array['20190101', '20190102'])
from access_log
where (dt='20190101' and page='detail.html') or
      (dt='20190102' and page='payment.html')
```

支持复杂表达式

针对更加复杂的查询，访问过'主页'或'搜索页'的用户，有多少访问了'详情页'？

将多个条件的结果集先或操作，再跟其它结果集做与操作。适用于于将多个物理埋点组装成一个逻辑埋点，再跟其它埋点做交集。

日期	UserID	PageId
20190101	100	index.html
20190101	102	search.html
20190101	100	detail.html
20190102	100	detail.html
20190102	102	search.html
20190102	100	search.html

```
select intersect_count(  
    user_id, page, array['首页|搜索页', '详情页'])  
from access_log
```

用户画像分析

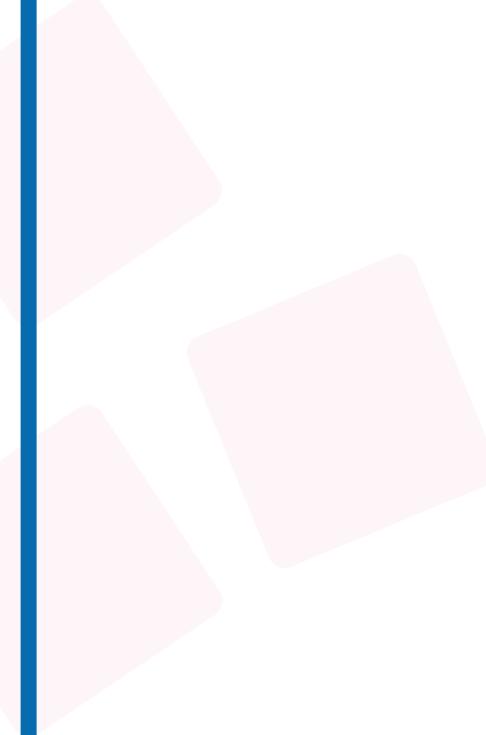
用户画像分析中需要通过标签进行用户的筛选；作为存储数字集合的最紧凑数据结构，Bitmap 常常被用在用户画像分析中。

使用 Kylin 做用户筛查时，通常需要将标签从列转行，将“标签类型”和“标签值”作为维度，将 User ID 作为 Bitmap 度量进行构建。

例如现在要分析，性别是男的、年龄是 90 后的、收入在 10-20 万区间的人有多少；通过 Kylin 这样查询即可！

UserID	TagName	TagValue
1000	性别	女
1000	年龄	80后
1000	收入	30-40万
1001	性别	男
1001	年龄	90后
1001	收入	10-20万

```
select intersect_count(  
    user_id, tag_value, array['男', '90后', '10-20万'])  
from user_profile  
where (tag_type='性别' and tag_value='男')  
    or (tag_type='年龄' and tag_value='90后')  
    or (tag_type='收入' and tag_value='10-20万')
```

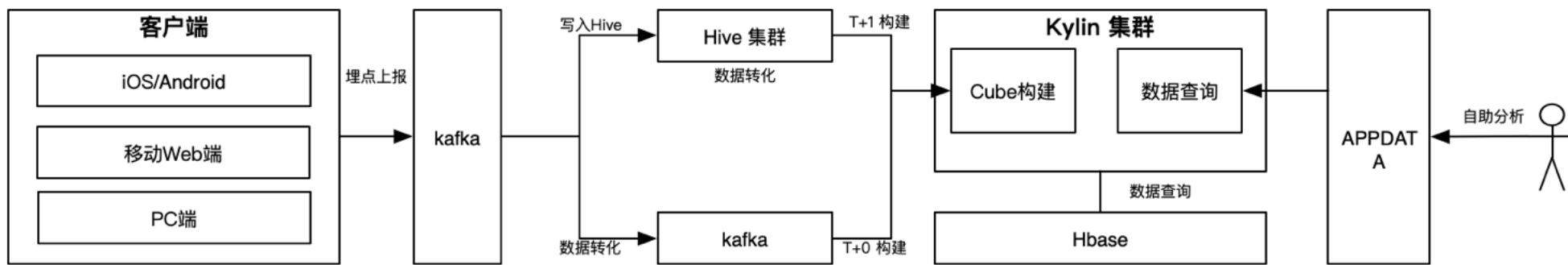


成功案例

滴滴出行 & 满帮集团

满帮 APPDATA

满帮集团，它是原运满满和货车帮合并后的集团，有 8 个手机 app，4 类埋点、上千个埋点值，收集了超过千亿条的用户行为日志。过去他们自己开发的分析平台各方面都不能满足业务需求，束缚了业务的发展。后来满帮集团迁移到了基于 Kylin 的分析平台，借助于 Kylin 的丰富功能，自研了名为 APPDATA 的一站式分析平台，极大地满足了业务对于数据分析的需求。





编辑行为

* 行为名称: 司机贷提款新-弹出确认借款框

埋点

页面-10005日志: 请选择PGN



自定义事件-10010日志: plugin_loan



_Loan_information_Enter_confirm...



plugin_loan



Loan_information_Enter_confirm...



H5-10004日志: 请选择Label



Web-10011日志埋点: 请选择Label



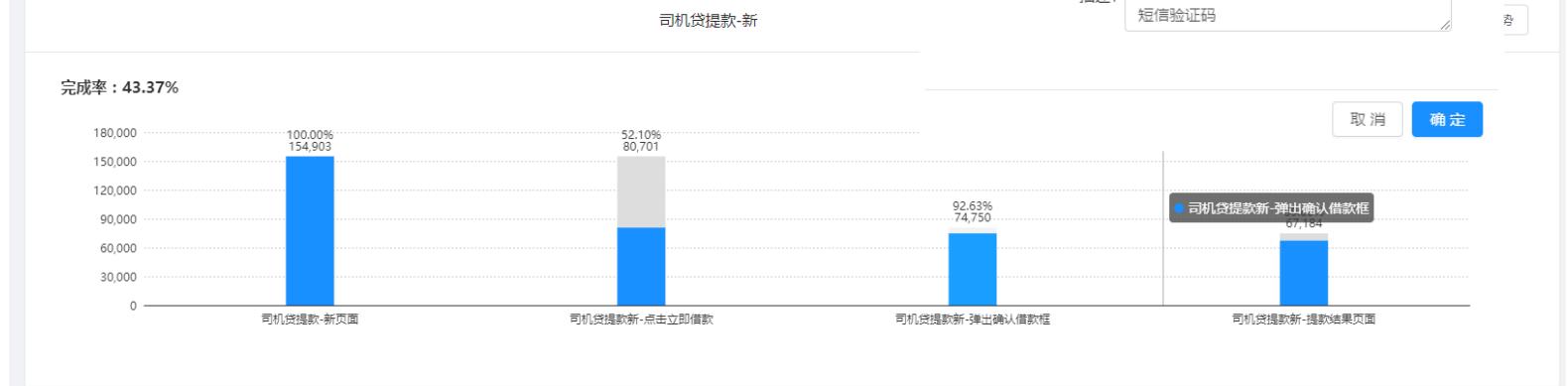
描述: 司机贷提款新-弹出确认借款框, 用于输入短信验证码

统计 ②
漏斗
筛选
导出

漏斗转化

全部漏斗 司机贷提款-新 管理

设备数 选择业务来源 2019-04-16 ~ 2019-05-15



导出数据

日期	总转化率	司机贷提款-新页面	司机贷提款-新页面	司机贷提款新-点击立即借款	司机贷提款新-点击立即借款	司机贷提款新-弹出确认借款框	司机贷提款新-弹出确认借款框	司机贷提款新-提款结果页面
汇总(2019-04-16--2019-05-15)	43.37%	154903	52.10%	80701	92.63%	74750	89.88%	67184
2019-04-16	32.44%	9332	42.88%	4002	89.61%	3586	84.41%	3027

满帮 APPDATA

基于 Kylin 的日／周／月留存分析报表，对于留存率异常情况，会通过颜色高亮显示

留存分析 ②

全部行为 ② 司机货提款新-弹出确认... ②

日留存 周留存 月留存 ②

时间	初始用户	回访用户									
		1天后	2天后	3天后	4天后	5天后	6天后	7天后	8天后	9天后	10天后
2019-05-15	4029										
2019-05-14	4070	334 8.21%									
2019-05-13	4002	399 9.97%	311 7.77%								
2019-05-12	3401	344 10.11%	258 7.59%	250 7.35%							
2019-05-11	3560	317 8.90%	299 8.40%	257 7.22%	250 7.02%						
2019-05-10	4074	343 8.42%	259 6.36%	284 6.97%	237 5.82%	240 5.89%					
2019-05-09	4332	396 9.14%	272 6.28%	260 6.00%	271 6.26%	253 5.84%	239 5.52%				
2019-05-08	4345	387 8.91%	342 7.87%	246 5.66%	236 5.43%	269 6.19%	235 5.41%	218 5.02%			
2019-05-07	4271	415 9.72%	333 7.80%	284 6.65%	255 5.97%	252 5.90%	242 5.67%	231 5.41%	239 5.60%		

满帮 APPDATA

满帮集团使用情况和运行
效率

千亿级数据量



每天近十亿数据增量，累计
10+T数据
8个客户端，覆盖全域流量数据
5个 cube, 16个维度

性能优良



Web 服务，响应 150+ 管理者/
产品/运营人员
API 服务，对接广告系统、活动
运营平台

配置简单



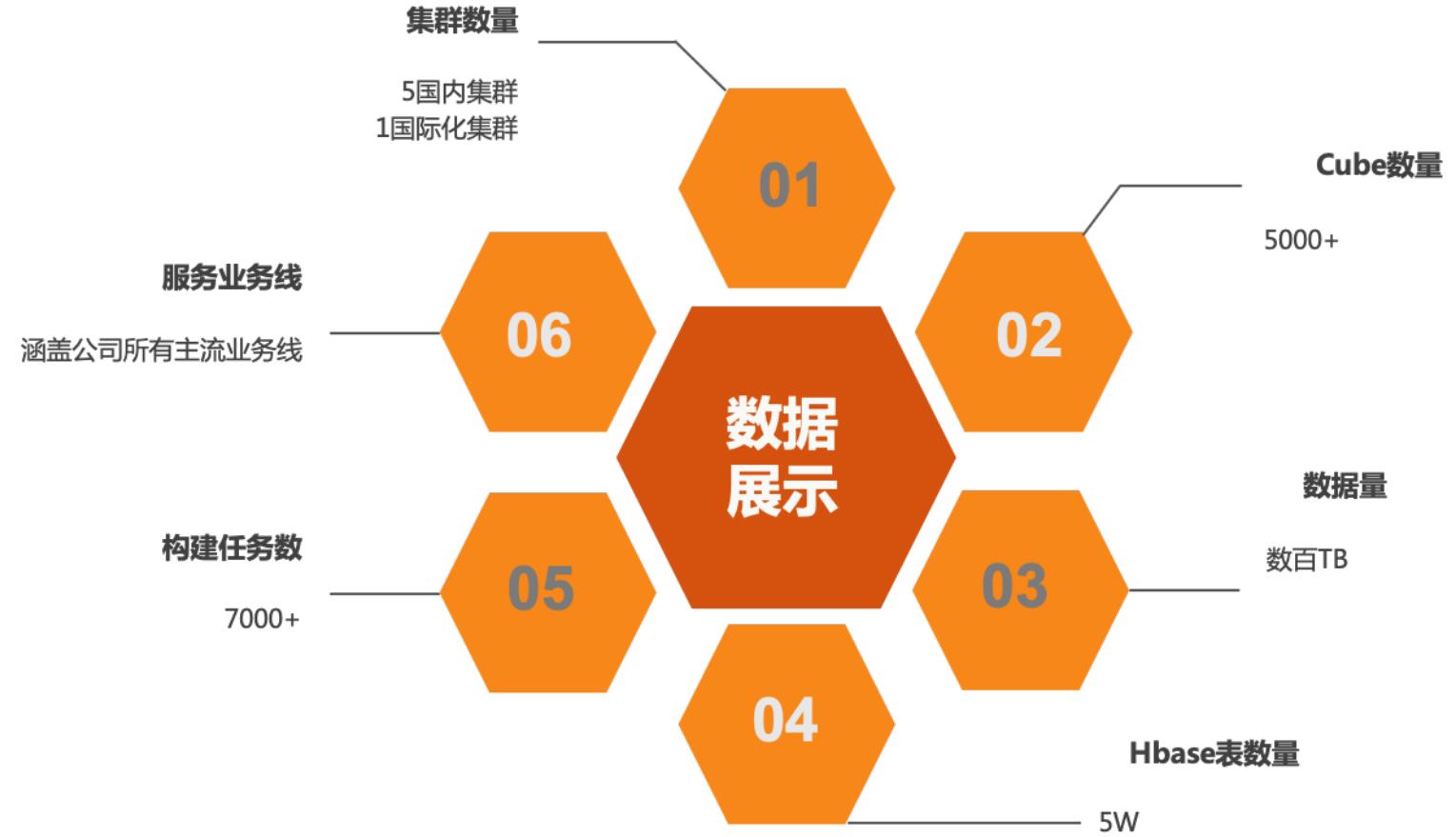
页面交互式操作，无需编程开发

亚秒级查询



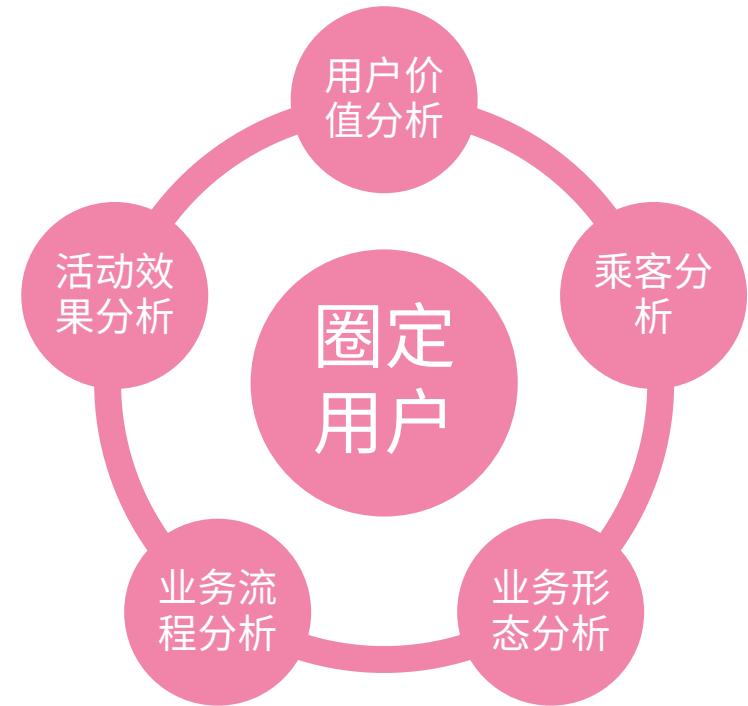
85%的查询是小于800毫秒
95%的查询小于3秒
99%的查询（复杂）小于7秒

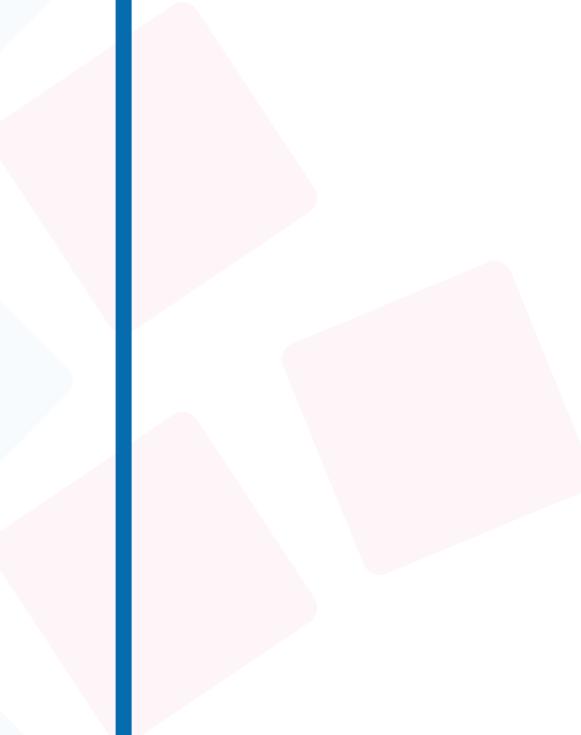
滴滴出行



滴滴出行

- 圈人发券（筛选用户群，定向选择发放优惠券）
- 乘客特征分析（跟画像系统集成，进一步确认圈人效果）
- 活动营销分析（观察促销活动后的效果）





总结

Kylin基于Bitmap的用户分析场景的优势

Kylin 为实现秒级精确去重引入了 Bitmap 做为用户集合的存储结构，通过扩展 SQL 聚合函数，Kylin 还支持对 Bitmap 的交集、先或再与以及查询明细等操作，可以非常巧妙地运用在用户行为和用户画像分析领域，相比于自己开发，有很多优势。



兼容SQL
易于使用，查询全部使用 SQL



性能优良
高性能高并发，大部分查询在秒级完成



配置简单
页面交互式操作，无需编程开发



稳定可靠
已在许多互联网用户如美团、滴滴、eBay 生产系统使用多年



欢迎加入Kylin Community

社区邮件组

- <http://kylin.apache.org/community>
- user@kylin.apache.org
- dev@kylin.apache.org

JIRA Issue

- <https://issues.apache.org/jira/projects/KYLIN>