# 基于Metis-Druid平台的移动数据分析服务

团队: 手机卫士服务端

讲师: 蒋冬临



# 提纲





1 Metis-druid 平台介绍

2 移动端日志解析与存储

3 多维实时查询与接入

#### Metis-druid 介绍 主要模块



每日处理的日志种类 150多种,,处理比 较繁琐,日志自动化 接入处理





数据分析的方式多样化 ,支持多维图表展示, 支持A|B数据对比,支 持top排名等功能分析





可以根据用户的阈值配置,对监控数据预警,同时可以对后续的一些行为进行预测



# 提纲



1 Metis-druid 平台介绍



2 移动端日志解析与存储

3 多维实时查询与接入

#### 移动端日志解析与存储



.....

移动端日志示例

"spec, stone, per, play". I god at the A company I HAT BE ! IF ---Manual . "symbox, app". I BE ARREST "ser", ARREST / 350 /W 'make' Section Sect. Companies 1. SERVI Small ! / DOM: AND 1. Although SHORT STREET, STREET W. A THE SEPT "Yesterlift, Auggest" FE CHATTABATA A management of the No. CHARLES NAME OF . ... SHOPPING THEFT !-I DES ADDES Salar, . .. I INCRESELS -CHES ASSES CHIEF STREET, NOTES "shee, marks": THE RE' WHERE CAST MARKET COMMO "phigh per"; "YOU JUST YOU PRODUCE BOARD OF BUILDING "the mile", FERRER "lever, nift, helicon". / STR SERIORS THE IT . I THE RESIDENCE WAR. I'V. ARREST MICHE L. SERBARET A we. TODA BREAK C. HIER. BERALE THE PERSON 'ww's 10000 Swine, Sor" (C September 2 2 2 SERVER DE 155.6F "Mit, Mot" Safer, page, M'-A REP AREAD THE REST COMP. TAXABLE STATE "refer and page of"-FEE AREES "physic version" / BW ANDS W 80 SHIP AT-FRE BURBO 100' 40 THE PERSON AND IN I WE BURGO No. Per 2 SHOWSHIM DETERM Secretary. I BE ACKS I BY MANAGE THE MAKE NO. WHAT AFFER ---"the artiset, a" ... Traffer, said, page, 18"-PER MARRIED SHAP BEIN SHARMS AT 1 80 A428 THERM M'-CHE SCAR Second Ballie "sim". · BY ABAT I BY BUFFIC "bull page, M"-Description of the Party SHOOM, M'S 1200 710-80 'exister') JET ANS "her person". Appropria 2.40% ..... "No setting A". FRE ATER "MAN, MANUF. AMOUNTS SHARL MET !! 1 000 BES 'bit, sworf', \$4000 Stramet y's FF AFCE "10" ... 7 7 50 2 SHINE Service See "teles". FF ASST Sec. A TIME BREEK (BREITABLE) / DES TON- MIC THE PARTY LEADING THE LA Seepen M". 40 - THE BEST BRUSHS To air divisit. Tight hat tax year of the fact of 2.85 ---"character". 1200 BEFES "purious norse, above", Table 44.2.8 Table 44.2.6" - 84.2.8" MIR. M. FREE COLD I THE STREAMSTERS, NAME OF "gray marks" Louising Colors 7 at "Mobile" CREW CREE "microsis" on Fidelik Broad Fill Server, Exertador", SERTERE "phylliane". FREE CREE THE PERSON NAMED IN version", I FRE CHE ! "more, dest" (b) SHOW MARKET - EARTH 361 CORF LENGE Tester page of "-\* FF 88580 Server Standard Select, Figuresia Safer, Ind. Aspt. Afric. "Bullet, balle". FRE FRAM. BLM 185 545 870 Server Americal reducedual". Talengal Fill. 4-7 MARCH SHOP, NO. 7 8 5 8 4 5 BO Server, Standard, Japan's Tightinger CHREST "NO JAME NO. **| 88 848870** Server Street, Scient, SHISHARD "VMP. ---'willer's I BE BOOK been tone returned \$50,000\$18 1 CAPE "NUMBER A" / NY ATON Louis gal / # Secure, P. INC ARCH 2 Post EW "man" ... / BT 8545 "spec, steer, pec, my", ... / BW JUNEFICKER / 2000 Z 14- BG "MYSIGNA", AT" o Named AND FRY ASSESS

#### 移动端日志解析与存储

```
360
www.360.cn
```

```
"用户属性1": "aaa",
      "用户属性2": "aaa",
      "用户属性3": "aaa",
      "用户属性4": "aaa",
      "用户属性5": "aaa",
花
      "用户属性6": "aaa",
      "用户行为":[
相
          "用户行为1": "abc",
机
         "用户时间": "dd",
          "用户行偏好": "abc"
测
试
         "用户行为1":"zxc",
数
          "用户时间": "qq",
         "用户行偏好": "fdg"
         "用户行为1": "gjh",
          "用户时间": "ew",
          "用户行偏好":"yu"
```

#### 扁平化格式的日志解析

按照字段将原有的复杂日志格式解析为扁平化、格式化数据。

プマロア**ノブ ・・** "用户属性4": "aaa"*,* "用户属性5": "aaa"*,* "用户属性6": "aaa",

"用户行为1":"abc",

展开部分: "用户时间": "dd", "用户行偏好": "abc"

#### 移动端日志解析与存储



"用户属性1": "aaa", "用户属性2": "aaa", "用户属性3": "aaa", "用户属性4": "aaa", "用户属性5": "aaa", "用户属性6": "aaa", "用户行为1": "abc", "用户时间": "dd", "用户行偏好": "abc"

"用户属性1": "aaa", "用户属性2": "aaa", "用户属性3": "aaa", "用户属性4": "aaa", "用户属性5": "aaa", "用户属性6": "aaa", "用户行为1": "zxc", "用户时间": "qq", "用户行偏好": "fdg"

2

"用户属性1": "aaa", "用户属性2": "aaa", "用户属性3": "aaa", "用户属性4": "aaa", "用户属性5": "aaa", "用户属性6": "aaa", "用户行为1": "gjh", "用户时间": "ew", "用户行偏好": "yu"

3

## 一条日志 展开为 多条日志

### 缺点:

- 1、日志量膨胀严重,公共部分信息冗余
- 2、解析后原本格式丢失



# Parquet 优势

1

\*压缩编码可以降低磁盘存储空间。由于同一列的数据类型是一样的,可以使用更高效的压缩编码进一步节约存储空间。

2

- \*可以跳过不符合条件的数据,只读取需要的数据
- ,降低IO数据量;
- \*只读取需要的列,能够获取更好的扫描性能。

3

\*支持多层嵌套的数据结构。既可以保持数据原有的层次,又可以满足多样化的查询需求。

#### 数据模型schema



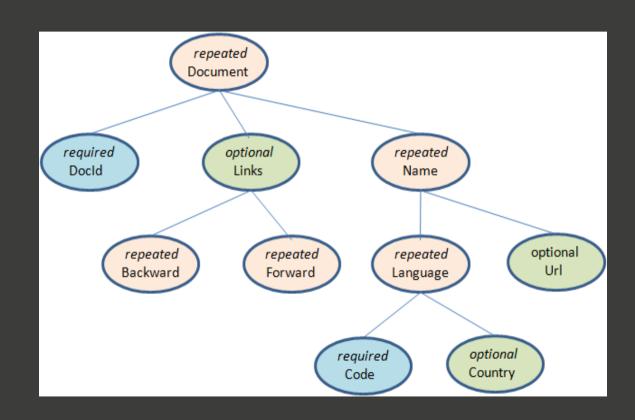
```
message Document {
                                  [1,1]
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
                                  [0,*]
    repeated int64 Forward;
  repeated group Name {
    repeated group Language {
      required string Code;
                                  [0,1]
      optional string Country;
    optional string Url;
```

```
DocId: 10
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-qb'
    Country: 'gb'
```



# 树状结构

- 1. 叶子节点存放基本类型数据
- 2. 没有Map、Array复杂数据结构



#### 数据模型schema



```
"age": 30.
 "company": {
   "title": "vvvv",
   "departName": "zzzzz"
 "jobs": [
      "projectName": "xxx develop",
     "projectType": "inner",
     "periods": {
       "startTime": "2016-10-01",
        "endTime": "2017-10-01"
      "projectName": "sms interception",
      "projectType": "to general customers",
      "periods": {
       "startTime": "2015-10-01",
       "endTime": "2022-10-01"
      }}]
```

```
Document {
required string name;
optional int32 age;
required group company {
  required string title;
  required string departName;
repeated group jobs {
  optional group periods {
    required string startTime;
    required string endTime;
  required string projectName;
  required string projectType;
```

#### 数据模型schema



```
message Document {
 required string name;
 optional int32 age;
                                                                                      Repeated
                                          required
                                                                       Optional
                                                        Required
 required group company {
    required string title;
    required string departName;
                                           required
                                                         required
  repeated group jobs {
                                                        departName
                                                                    optional
                                                                                required
                                                                                             required
    optional group periods {
                                                                                string
                                                                    group
                                                                               proiectName
                                                                                            projectType
                                                                    periods
       required string startTime;
       required string endTime;
    required string projectName;
                                                              required
                                                                           required
                                                              string
                                                                            string
    required string projectType;
                                                              startTime
                                                                           endTime
```

#### Striping/Assembly算法



#### 在该算法中列的每一个值都包含三部分

- 1. Value
- 2. repetition level
  - 在写入的时候该值等于它和前面的值在哪一层节点是不共享的
  - 在读取的时候推导出哪一层上需要创建一个新的节点
- 3. definition level:对于NULL来说,路径p上有多少字段可以是不存在(例如在文档定义中是optional或repeated,而不是required),然而实际却存在的

#### **Repetition and definition levels**



DocId: 10 Links Forward: 20 Forward: 40 Forward: 60 Name Language Code: Country: 'us' Language Code: Url: 'http://A' Name Url: 'http://B' Name Language Code: Country: 'qb'

: common prefix

r: repetition level

d: definition level

#### Name.Language.Code

 value
 r
 d

 en-us
 0
 2

 en
 2
 2

 NULL
 1
 1

 en-gb
 1
 2

doc<sub>1</sub>.Name<sub>1</sub>.Language<sub>1</sub>.Code: 'en-us'

doc<sub>1</sub>.Name<sub>1</sub>.Language<sub>2</sub>.Code: 'en'

doc<sub>1</sub>.Name<sub>2</sub>

doc<sub>1</sub>.Name<sub>3</sub>.Language<sub>1</sub>.Code: 'en-gb'

#### 最终数据格式



Docld

value	r	d
10	0	0
20	0	0

Name.Url

value	r	d
http://A	0	2
http://B	1	2
NULL	1	1
http://C	0	2

Links.Forward

value	r	d
20	0	2
40	1	2
60	1	2
80	0	2

Links.Backward

value	r	d
NULL	0	1
10	0	2
30	1	2

Name.Language.Code

value	r	d
en-us	0	2
en	2	2
NULL	1	1
en-gb	1	2
NULL	0	1

Name.Language.Country

value	r	d
us	0	3
NULL	2	2
NULL	1	1
gb	1	3
NULL	0	1

列式存储格式

### 应用数据对比



数据量大小	业务1数据	业务2数据
原始日志数据	42.62G	688.16G
原始数据Gz压缩存储	5.17G	99.49G
原始数据 Parquet+gz	3.02G	16.56G
原始日志压缩比例	14.11倍	42.21倍
相比Gzip 压缩比例	1.7倍	6.00倍

## 应用数据对比



打点数据大小	数据量	统计PV	统计PV    某一字段去重统计    投影某两列	
扁平化行式	548.46M	13.917 s	31.016 s	50.177 s
parquet	26.39M	7.053 s	10.498 s	8.448 s
比例	20.78倍	1.97倍	<b>3.10</b> 倍	5.9倍

# 提纲



1 Metis BI 介绍

2 移动端日志解析与存储





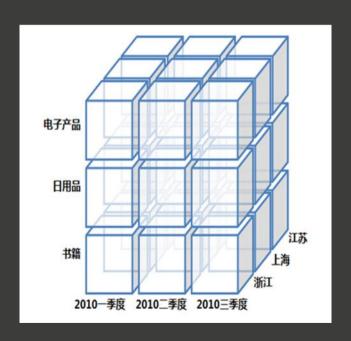
#### **OLAP(Online Analytical Processing)**

- > 联机分析处理
- ▶ 快速、灵活地进行 大数据量的**多维度** 实时查询。





#### 数据立方体



#### Druid主要解决问题场景



- ▶ 百亿千亿级别日志,独立访客(UV)的统计
- > 多维度多样化的留存率分析
- ▶ top-k elements,统计top APP访问排行
- ➤ Range Query,统计一段时间范围的网站访问量(PV,UV)
- > Group by dimensions 分析

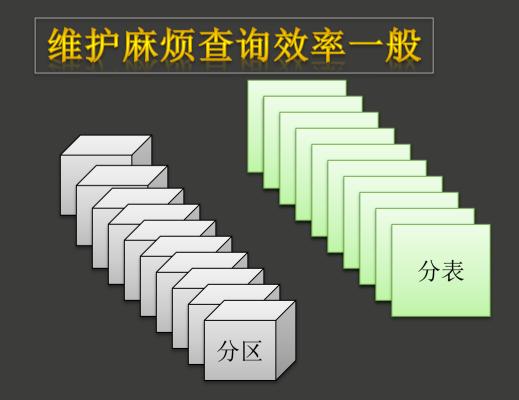


>多维查询下,如果多达20维以上。。



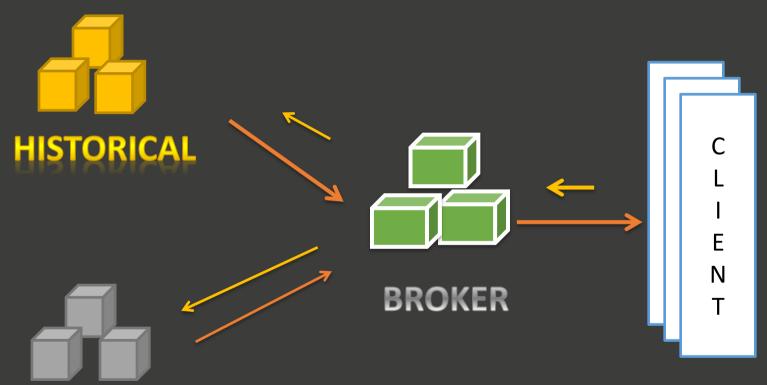
查询慢、查不了

怎么办?



#### Druid主要解决问题场景

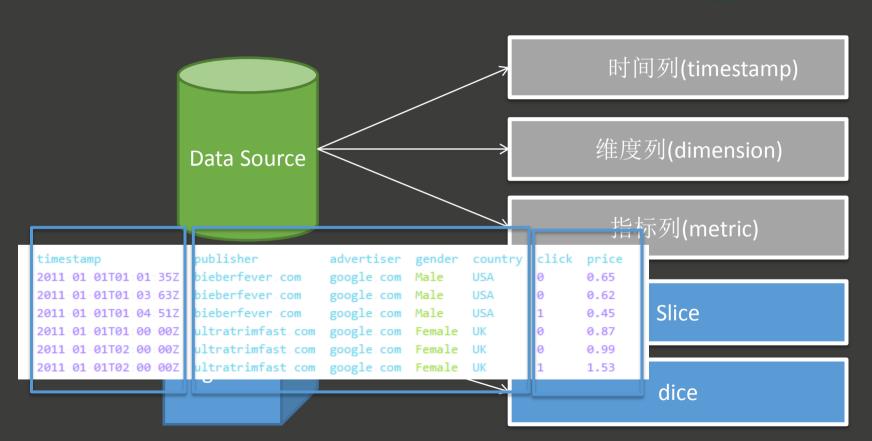




- ■使用Druid 使用Druid ,查有的的一种的一种, 有力, 有力, 有力, 有一种, 有一种。
- □查询多个 维度的top 排序可 级内 返回;

REALTIME





# **Druid roll-up**



timestamp	publisher	advertiser	gender	country	click	price
2011-01-01T01:01:35Z	bieberfever.com	google.com	Male	USA	0	0.65
2011-01-01T01:03:63Z	bieberfever.com	google.com	Male	USA	0	0.62
2011-01-01T01:04:51Z	bieberfever.com	google.com	Male	USA	1	0.45
2011-01-01T01:00:00Z	ultratrimfast.com	google.com	Female	UK	0	0.87
2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Female	UK	0	0.99
2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Female	. UK	1	1.53



	timestamp	publisher	advertiser	gender	country	impressions	s click	ks reveni
	2011-01-01T01:00:00Z	ultratrimfast.com	google.con	n Male	USA	1800	25	15.70
ı	2011-01-01T01:00:00Z	bieberfever.com	google.com	n Male	USA	2912	42	29.18
ì								
	2011-01-01T02:00:00Z	ultratrimfast.com	google.con	n Male	UK 1	1953	17	17.31
ı	2011-01-01T02:00:007	bieberfever.com	google.com	n Male	UK 3	3194	170	34.01

segment1

segment2



- Druid的查询使用HTTP REST风格的请求,发送给查询节点(Broker -> Historical, Realtime)
- Druid查询主要通过JSON文件指定维度和聚合

Druid的聚合查询主要有三种形式:

- 1. Timeseries
- 2. TopN
- 3. GroupBy



```
"queryType":"timeseries",
 "dataSource": "dap yy",
 "granularity": {"type": "period", "period": "P1D", "timeZone":
"Asia/Shanghai"},
 "threshold": 20,
 "aggregations": [
   "type": "longSum", "name": "active user", "fieldName":
"active user"
   "type": "longSum", "name": "active user1", "fieldName":
"active user1"
 "intervals": [
    "2017-01-01/2017-01-03"
```

Timeseries输出每个时间粒度内指定条件的统计信息,通过filter来指定过滤条件,通过aggregations和PostAggregations指定聚合方式。



```
"queryType":"topN",
"dataSource": "dap_xx",
"granularity": {"type": "period", "period": "P1D", "timeZone": "Asia/Shanghai"},
"dimension": "channel",
"threshold": 20,
"metric": "active user",
"aggregations": [
  "type": "longSum", "name": "active user", "fieldName": "active user"
  "type": "longSum","name": "active_user1","fieldName": "active_user1"
"intervals": [
   "2017-01-01/2017-01-03"
```

- 1. TopN查询返回的是根据某一维度进行group by后再排序,返回结果集
- 2. TopN的查询是近似查询,每个节点会返回topK(max(1000, threshold)),实际使用上,以top1000为例,前900个的结果项是精确的,而后100个则不能保证精确性



```
"queryType": "groupBy",
"dataSource": "dap xx",
"granularity": {"type": "period", "period": "P1D", "timeZone": "Asia/Shanghai"},
"dimensions": ["channel","province","brand"],
"limitSpec": { "type": "default", "limit": 10, "columns": [{"dimension": "active",
"direction": "descending", "dimensionOrder": "numeric" }]},
"aggregations": [
  {"type": "longSum", "name": "active", "fieldName": "active user"},
  {"type": "longSum", "name": "active1", "fieldName": "active user1"},
  {"type": "longSum", "name": "active3", "fieldName": "active user3"},
  {"type": "longSum", "name": "active7", "fieldName": "active user7"},
  {"type": "longSum", "name": "active30", "fieldName": "active user30"}
 "filter": {
  "tvpe": "not".
  "field": {"type": "in", "dimension": "channel", "values": ["\"other\"",9999]}
 "intervals": ["2017-01-09/2017-01-10"]
```

GroupBy类似于SQL中的group by 操作, 能对指定的多维度进行分组。

谢 谢!