

# 系统设计

C01 研发部-统计平台项目组

2013 年 11 月

## 目录

系统设计 .....	1
C01 研发部-统计平台项目组 .....	1
2013 年 11 月 .....	1
1、日志格式 .....	2
1.1、人数 人次(该示例已插入数据库配置表) .....	3
1.2、人数 人次 数量 .....	4
1.3、人数 人次 数量 1 数量 2 .....	4
1.4、人数 人次 数量 等级分布(该示例已插入数据库配置表) .....	4
1.5、人数人次各子项的人数人次 .....	5
1.6、人数 人次 各子项的人数人次数量 等级分布 .....	6
1.7、最大值 累加值分布的人数人次 .....	6
2、数据库设计 .....	6
2.1、游戏基本信息表 t_game_info .....	6
2.2、游戏平台区服映射表 t_gpzs_info .....	7
2.3、树结构表 t_web_tree .....	7
2.4、reportId 映射表 t_report_info .....	8
2.5、数据项映射表 t_data_info .....	9
2.6、导航表 t_web_navi .....	10
3、模块设计 .....	11
3.1、Stat-logger .....	12
3.2、Stat-client .....	12
3.3、Stat-proxy .....	12
3.4、Stat-server .....	12

3.5、stat-center .....	12
3.6、Db-server .....	13
3.6.1、分钟数据的计算 .....	13
3.6.2、配置项的入库 .....	13
3.6.2、数据计算结果的入库 .....	13
3.7、config-server .....	13
3.8、mapreduce .....	13
3.9、mapreduce 计算结果入库 .....	13

## 1、日志格式

字段名	含义
hip	发送数据的客户端 IP
stid	统计项名称，比如游戏输出
sstid	子统计项名称，比如撞球对抗赛、黄金矿工
gid	游戏 ID
zid	区 ID
sid	服 ID
pid	平台 ID -1 表示全平台，默认淘米平台 ID 为 1
ts	时间戳
acid	帐号(米米号)
plid	角色标识
key1=value1, key2=value2,...	自定义 key-value 对
op	对自定义 key 进行的操作

op 取值	说明
计算类型	
sum(key)	对指定 key 的 value 求和
max(key)	对指定 key 的 value 求最大值
set(key)	对指定 key 的 value 做 set 做操作
item(key)	对指定 key 的 value 计算出对应的人数人次
item_sum(key1, key2)	按照 key1 分类，对 key2 的 value 字段求和
item_max(key1, key2)	按照 key1 分类，对 key2 的 value 字段求最大值
item_set(key1, key2)	按照 key1 分类，对 key2 的 value 字段做 set 操作，即取每天最后一个值
sum_distr(key)	需要依赖于区间分布，然后计算出对 key 的 value 字段求和后属于哪个区间的分布

max\_distr(key)            需要依赖于区间分布，然后计算出对 key 的 value 字段求最大值后属于哪个区间的分布

set\_distr(key)            对 key 的 value 字段做 set 操作后做分布

ip\_distr                    ip 分布，需要根据 IP 地址查出对应的地区

对每一条日志，都会默认计算出人数人次；

每条日志在后台计算的时候都会扩充成至少三条日志进行计算，主要是对 gid、zid、sid、pid 进行扩充

比如

Gid(游戏 ID)	Zid(区 ID)	Sid(服 ID)	Pid(平台 ID)	备注
15	2	4	1	原始日志, 游戏 ID 为 15 的游戏在淘米平台上的 2 区 4 服
15	-1	-1	1	扩充出来的日志, 游戏 ID 为 15 的游戏在淘米平台上的全区全服数据
15	-1	-1	-1	扩充出来的日志, 游戏 ID 为 15 的游戏在全平台全区全服的数据

下面是具体的日志示例：

## 1.1、人数 人次(该示例已插入数据库配置表)

**原始的日志格式：**

\_stid\_=领地争夺战 \_sstid\_=点击传送舱装置 acid=具体的米米号 plid=角色标识(如果需要按角色查看，不需要按角色查看时，为空即可)——可计算出在领地争夺战中点击传送舱装置的人数以及人次

**转发给实时计算模块的日志格式：**

\_stid\_=领地争夺战 \_sstid\_=点击传送舱装置 acid=米米号 plid=角色标识 op=count(1)

**原始的日志格式：**

\_stid\_=领地争夺战 \_sstid\_=成功报名 acid=具体的米米号 plid=角色标识(如果需要按角色查看，不需要按角色查看时，为空即可)——可计算出在领地争夺战中成功报名的人数以及人次

**转发给实时计算模块的日志格式：**

\_stid\_=领地争夺战 \_sstid\_=成功报名 acid=米米号 plid=角色表示 op=count(1)

## 1.2、人数 人次 数量

原始的日志格式:

\_stid\_=游戏输出 sstid=撞球对抗赛 acid=米米号 1 plid=角色标识 1 赛尔豆=17 op=sum:赛尔豆

\_stid\_=游戏输出 sstid=撞球对抗赛 acid=米米号 2 plid=角色标识 2 赛尔豆=27 op=sum:赛尔豆

——可计算出撞球对抗赛的人数人次以及输出的赛尔豆总和

转发给实时计算模块的日志格式:

\_stid\_=游戏输出 \_sstid\_=撞球对抗赛 acid=米米号 1 plid=角色标识 1 op=count:1

\_stid\_=游戏输出 \_sstid\_=撞球对抗赛 acid=米米号 1 plid=角色标识 1 赛尔豆=17 op=sum:赛尔豆

\_stid\_=游戏输出 \_sstid\_=撞球对抗赛 acid=米米号 2 plid=角色标识 2 op=count:1

\_stid\_=游戏输出 \_sstid\_=撞球对抗赛 acid=米米号 2 plid=角色标识 2 赛尔豆=27 op=sum:赛尔豆

## 1.3、人数 人次 数量 1 数量 2

原始的日志格式:

\_stid\_=购买家具类商品 \_sstid\_=皇冠猪猪屋 acid=具体的米米号 plid=角色标识 购买数量=10 花费金额=180 op=sum:购买数量|sum:花费金额

——可计算出购买皇冠猪猪屋的人数人次、购买总数量、金额总和

如果要分别统计 VIP 和非 VIP 用户购买的数量, \_stid\_ 分别填成 vip 用户购买家具类商品和非 VIP 用户购买家具类商品

转发给实时计算模块的日志格式:

\_stid\_=购买家具类商品 \_sstid\_=皇冠猪猪屋 acid=具体的米米号 plid=角色标识 op=count:1

\_stid\_=购买家具类商品 \_sstid\_=皇冠猪猪屋 acid=具体的米米号 plid=角色标识 购买数量=10 op=sum:购买数量

\_stid\_=购买家具类商品 \_sstid\_=皇冠猪猪屋 acid=具体的米米号 plid=角色标识 花费金额=180 op= sum:花费金额

## 1.4、人数 人次 数量 等级分布(该示例已插入数据库配置表)

原始的日志格式:

\_stid\_=运营活动 \_sstid\_=宇宙大漫游 acid=具体的米米号 plid=角色标识 经验值=120 等级='11-20' 18 op=sum:经验值|item:等级

——可计算出参加宇宙大漫游活动的人数人次以及输出的经验值,还有参加该活动的各等级用户数

转发给实时计算模块的日志格式:

\_stid\_=运营活动 \_sstid\_=宇宙大漫游 acid=具体的米米号 plid=角色标识 op=count:1

\_stid\_=运营活动 \_sstad=宇宙大漫游 acid=具体的米米号 plid=角色标识 经验值=120  
op=sum:经验值

\_stid\_=运营活动 \_sstad=宇宙大漫游 acid=具体的米米号 plid=角色标识 等级= 18 op=  
count:1

(可以求出 item 的人次, 遇到 op=count 的情况, 前面所有的都是 key, 如果 op!=count, 那么  
只有最后一个字段不是 key)

**原始的日志格式:**

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 1 plid=角色标识 1 赛尔豆=17 op=sum:赛尔  
豆

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 2 plid=角色标识 2 赛尔豆=27 op=sum:赛尔  
豆

——可计算出保护导航仪的人数人次以及输出的赛尔豆总和

**转发给实时计算模块的日志格式:**

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 1 plid=角色标识 1 op=count:1

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 1 plid=角色标识 1 赛尔豆=17 op=sum:赛尔  
豆

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 2 plid=角色标识 2 op=count:1

\_stid\_=游戏输出 sstad=保护导航仪 acid=米米号 2 plid=角色标识 2 赛尔豆=27 op=sum:赛尔  
豆

## 1.5、人数人次各子项的人数人次

**原始的日志格式:**

\_stid\_=新手任务 \_sstad=制作物品 acid=米米号 plid=角色标识 关卡=第 1 关 op=item:关卡

\_stid\_=新手任务 \_sstad=制作物品 acid=米米号 plid=角色标识 关卡=第 2 关 op=item:关卡

\_stid\_=新手任务 \_sstad=制作物品 acid=米米号 plid=角色标识 关卡=第 3 关 op=item:关卡

\_stid\_=新手任务 \_sstad=制作物品 acid=米米号 plid=角色标识 关卡=第 4 关 op=item:关卡

\_stid\_=新手任务 \_sstad=制作物品 acid=米米号 plid=角色标识 关卡=第 5 关 op=item:关卡

——可计算出各关卡的人数人次

也可以通过将 \_stid\_=新手任务-制作物品 \_sstad=第 1 关的方式, 按照第 1 种人数人次的方  
式做

**原始的日志格式:**

\_stid\_=战豆分布 \_sstad=战豆产出 acid=米米号 plid=角色标识 产出数量=' 1-10' op=item:  
产出数量

\_stid\_=战豆分布 \_sstad=战豆产出 acid=米米号 plid=角色标识 产出数量=' 11-20' op=item:  
产出数量

\_stid\_=战豆分布 \_sstad=战豆产出 acid=米米号 plid=角色标识 产出数量=' 21-50' op=item:  
产出数量

\_stid\_=战豆分布 \_sstad=战豆产出 acid=米米号 plid=角色标识 产出数量=' 51 以上' op=item:  
产出数量

——可统计出战豆产出的区间分布人数

## 1.6、人数 人次 各子项的人数人次数 等级分布

原始的日志格式:

\_stid\_=购买道具 \_sstad\_=白甲机动战士 acid=米米号 plid=角色标识 职业=熊猫 数量=15 价格=150 等级=15 op=sum:数量|item:职业|item\_sum:职业,数量|item\_sum:职业,价格|item\_set\_distr:职业,等级

\_stid\_=购买道具 \_sstad\_=白甲机动战士 acid=米米号 plid=角色标识 职业=兔子 数量=15 价格=150 等级=18 op= sum:数量|item:职业|item\_sum:职业,数量|item\_sum:职业,价格|item\_set\_distr:职业,等级

\_stid\_=购买道具 \_sstad\_=白甲机动战士 acid=米米号 plid=角色标识 职业=猴子 数量=15 价格=150 等级=24 op= sum:数量|item:职业|item\_sum:职业,数量|item\_sum:职业,价格|item\_set\_distr:职业,等级

——可计算出购买各个道具的人数人次 以及各个职业购买的人数人次,数量总和,价格总和以及购买该道具的各职业用户等级分布

## 1.7、最大值 累加值分布的人数人次

原始的日志格式

\_stid\_=战豆分布 \_sstad\_=战斗消耗 acid=米米号 plid=角色标识 消耗数量=3 op=dsitr\_sum:消耗数量|distr\_max:消耗数量

\_stid\_=战豆分布 \_sstad\_=战斗消耗 acid=米米号 plid=角色标识 消耗数量=16 op=dsitr\_sum:消耗数量,区间|distr\_max:消耗数量,区间

——可计算出消耗的战豆分布 分最大和累积

# 2、数据库设计

## 2.1、游戏基本信息表 t\_game\_info

字段名	类型	备注
game_id	int	主键字段 游戏 ID 与 account 平台保持一致
game_name	char(64)	唯一键字段 游戏名称
game_type	enum	游戏类型页游(webgame) 手游 (mobilegame) 端游 (clientgame) 测试(test)
auth_id	varchar(32)	游戏权限 ID, 由权限系统分配
status	tinyint	状态字段 0:未使用 1: 已使用 2: 删除

## 2.2、游戏平台区服映射表 t\_gpzs\_info

字段名	类型	备注
gpzs_id	bigint	自增主键
game_id	int	唯一键字段 游戏 ID 与 t_game_info 表的 game_id 一致
platform_id	int	唯一键字段 平台 ID, 如淘米平台规定为 1
zone_id	int	唯一键字段 区 ID, 多区多服游戏需要使用,默认全区全服为-1
server_id	int	唯一键字段 服 ID, 多区多服游戏需要使用,默认全区全服为-1
gpzs_name	char(64)	平台区服名字
status	tinyint	状态字段 0: 正常 1: 下架

## 2.3、树结构表 t\_web\_tree

字段名	类型	备注
node_id	int	自增主键
node_name	char(128)	节点名字, 默认取值 stid 或 sstid 支持修改
game_id	int	唯一键字段 游戏 id
parent_id	int	父节点 ID
is_leaf	tinyint	是否叶子节点 1:叶子节点 0: 非叶子节点
is_basic	tinyint	是否基础统计项 1:基础统计项 0: 非基础统计项
hide	tinyint	是否隐藏 0:显示 1: 隐藏
stid	char(64)	唯一键字段 统计项名称
sstid	char(64)	唯一键字段 子统计项名称

>>比如一条在线日志和自定义的赛尔豆产出日志

```
>>_hip_=10.1.1.184 _stid=_olcnt_ _sstid=_olcnt_ _gid_=13 _zid_=-1 _sid_=-1 _pid_=1
    _ts_=1383705667 _acid_=-1 _plid_=-1 _olcnt_=0 _op_=max:_olcnt_
>>_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=撞球对抗赛 _gid_=2 _zid_=-1 _sid_=-1 _pid_=1
    _ts_=1383705667 _acid_=47159775 _plid_=-1 赛尔豆=100 _op_=sum:赛尔豆
>>_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=划船比赛 _gid_=2 _zid_=-1 _sid_=-1 _pid_=1
    _ts_=1383705667 _acid_=47159775 _plid_=-1 赛尔豆=100 _op_=sum:赛尔豆
>>_hip_=10.1.1.182 _stid_=每周任务(接收) _sstid_=幸福的使命 _gid_=5 _zid_=-1 _sid_=-1
    _pid_=1 _ts_=13837896 _acid_=34545221 _plid_=-1 lv='1-5' _op_=item:lv
```

```
>>_hip_=10.1.1.182 _stid_=每周任务(接收) _sstid_=幸福的使命 _gid_=5 _zid_=-1 _sid_=-1
_pid_=1 _ts_=13837896 _acid_=13545221 _plid_=-1 lv='6-10' _op_=item:lv
>>_hip_=10.1.1.182 _stid_=每周任务(接收) _sstid_=幸福的使命 _gid_=5 _zid_=-1 _sid_=-1
_pid_=1 _ts_=13837896 _acid_=78545221 _plid_=-1 lv='11-20' _op_=item:lv
>>_hip_=10.1.1.182 _stid_=每周任务(接收) _sstid_=幸福的使命 _gid_=5 _zid_=-1 _sid_=-1
_pid_=1 _ts_=13837896 _acid_=5654345221 _plid_=-1 lv='21 以上' _op_=item:lv
```

node_id	node_name	game_id	parent_id	is_leaf	is_basic	hide	stid	sstid
1	_olcnt_	13	0	0	1	0	_olcnt_	'空字符串'
2	_olcnt	13	1	1	1	0	_olcnt_	_olcnt_
3	游戏输出	2	0	0	0	0	游戏输出	'空字符串'
4	撞球对抗赛	2	3	1	0	0	游戏输出	撞球对抗赛
5	划船比赛	2	3	1	0	0	游戏输出	划船比赛
6	每周任务(接收)	5	0	0	0	0	每周任务(接收)	
7	幸福的使命	5	6	1	0	0	每周任务(接收)	幸福的使命

## 2.4、reportId 映射表 t\_report\_info

字段名	类型	备注
report_id	bigint	自增 主键
report_name	char(64)	统计项名称 默认取 stid-sstid
game_id	int	唯一键字段 游戏 id
stid	char(64)	唯一键字段 统计项名称
sstid	char(64)	唯一键字段 子统计项名称
op_fields	char(64)	唯一键字段 操作的字段
op_type	enum	唯一键字段 操作类型 人次(count) 人数(ucount) 求和(sum) 最大(max) 覆盖(set) IP 分布(ip_distr) 等级分布(distr_max,distr_sum,distr_set)
is_multi	tinyint	0- 非 multi 类型, 1- multi 类型

比如对于上面的三条日志会产生如下的记录

report_id	report_name	game_id	stid	sstid	op_fields	op_type	
1	_olcnt_-_olcnt_	13	_olcnt_	_olcnt_	_olcnt_	max	
2	游戏输出-撞球	2	游戏输	撞球对	空字段	Count	



	对抗赛		出	抗赛	串		
3	游戏输出-撞球对抗赛	2	游 戏 输 出	撞 球 对 抗赛	空 字 段 串	Ucount	
4	游戏输出-撞球对抗赛	2	游 戏 输 出	撞 球 对 抗赛	赛尔豆	Sum	
5	游戏输出-划船比赛	2	游 戏 输 出	划 船 比 赛	空 字 段 串	Count	
6	游戏输出-划船比赛	2	游 戏 输 出	划 船 比 赛	空 字 段 串	Ucount	
7	游戏输出-划船比赛	2	游 戏 输 出	划 船 比 赛	赛尔豆	Sum	
8	每周任务(接收)-幸福的使命	5	每 周 任 务(接收)	幸 福 的 使命	空 字 符 串	Count	
9	每周任务(接收)-幸福的使命	5	每 周 任 务(接收)	幸 福 的 使命	空 字 符 串	Ucount	
10	每周任务(接收)-幸福的使命	5	每 周 任 务(接收)	幸 福 的 使命	lv	Count	
11	每周任务(接收)-幸福的使命	5	每 周 任 务(接收)	幸 福 的 使命	lv	ucount	

## 2.5、数据项映射表 t\_data\_info

字段名	类型	备注
data_id	bigint	自增主键
data_name	char(64)	数据项名称 默认取值为 range - datatype
r_id	Bigint	唯一键字段 代表是统计项 report_id 还是加工项 result_id
type	Enum	唯一键字段 统计项(report) 加工项(result)
range	Char(64)	唯一键字段 范围字段
display_order	Int	排序字段,显示时用
hide	tinyint	是否隐藏 0:显示 1:隐藏
shtash	int	stid 的 hash 值,用于分库分表

比如上面的日志入库记录如下

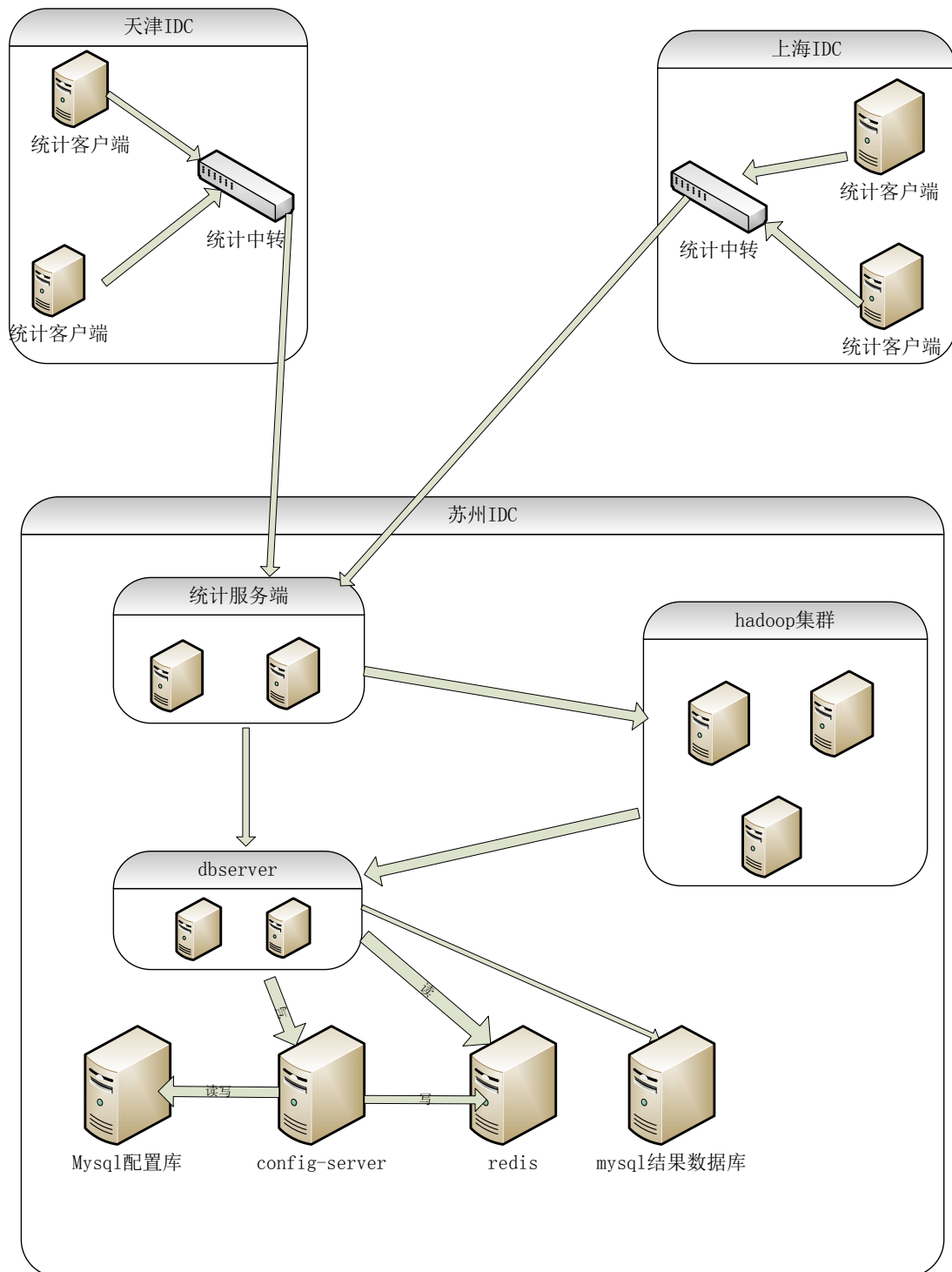
Data_id	Data_name	R_id	type	range	Display_order	hide	sthash
1	在线人数	1	Report	空 字 符	0	0	13

				串			
2	人次	2	Report	空字符串	0	0	15
3	人数	3	Report	空字符串	0	0	15
4	求和	4	Report	空字符串	0	0	15
5	人次	5	Report	空字符串	0	0	15
6	人数	6	Report	空字符串	0	0	15
7	求和	7	Report	空字符串	0	0	15
8	人次	8	Report	空字符串	0	0	19
9	人数	9	Report	空字符串	0	0	19
10	人次	10	Report	1-5	0	0	19
11	人次	10	Report	6-10	0	0	19
12	人次	10	Report	11-20	0	0	19
13	人次	10	Report	20 以上	0	0	19
14	人数	11	Report	1-5	0	0	19
15	人数	11	Report	6-10	0	0	19
16	人数	11	Report	11-20	0	0	19
17	人数	11	report	20 以上	0	0	19

## 2.6、导航表 t\_web\_navi

字段名	类型	备注
navi_id	int	自增主键
navi_name	varchar(128)	导航名字
navi_key	varchar(128)	导航键，用于加载静态资源
navi_url	varchar(128)	导航地址
parent_id	int	父导航 ID
auth_id	varchar(32)	导航权限 ID，由权限系统分配
level	tinyint	导航层级
display_order	int	导航显示顺序，值越小越靠前，同层级有效
status	tinyint	显示状态：1-显示，0-不显示

### 3、模块设计



### 3.1、Stat-logger

提供给项目部使用，用于收集统计日志,落到本地

### 3.2、Stat-client

将项目部落在本地的统计日志发送给统计中转  
具体实现细节待补充

### 3.3、Stat-proxy

各个 IDC 部署一个中转，负责转发统计日志给服务端  
具体实现细节待补充

### 3.4、Stat-server

统计服务端，主要功能分为两块

- 1、将接收到的每条日志写本地文件，由专门的脚本定时上传到 hadoop 集群
- 2、根据每条日志的 OP 字段对日志进行拆分，默认都会拆分出一条人次的日志发给 dbserver 模块

比如下面一条日志：

```
_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=撞球对抗赛 _gid_=2 _zid_=-1 _sid_=-1  
_pid_=1 _ts_=1383705667 _acid_=47159775 _plid_=-1 赛尔豆=100 lv=23 _op_=sum:赛  
尔豆|set_distr:lv
```

会被拆成下面三条条日志发给 db-server

```
_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=撞球对抗赛 _gid_=2 _zid_=-1 _sid_=-1  
_pid_=1 _ts_=1383705667 _acid_=47159775 _plid_=-1 _op_=count //统计人次  
_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=撞球对抗赛 _gid_=2 _zid_=-1 _sid_=-1  
_pid_=1 _ts_=1383705667 _acid_=47159775 _plid_=-1 赛尔豆=100 _op_=sum:赛尔豆  
_hip_=10.1.1.60 _stid_=游戏输出 _sstid_=撞球对抗赛 _gid_=2 _zid_=-1 _sid_=-1  
_pid_=1 _ts_=1383705667 _acid_=47159775 _plid_=-1 lv=23 _op_=set_distr:lv
```

具体实现细节待补充

### 3.5、stat-center

暂时未实现，将来用于控制统计系统中各个模块的

## 3.6、Db-server

Dbserver 模块负责实时数据也即分钟数据的计算、配置项的入库和数据计算结果的入库

### 3.6.1、分钟数据的计算

待补充

### 3.6.2、配置项的入库

先从 redis 查询配置项是否存在，

如果已经存在，忽略

如果不存在，将项目配置数据传给 config-server, config-server 会直接插入数据库相关配置表，然后更新到缓存 redis 中

细节待补充

### 3.6.2、数据计算结果的入库

待补充

## 3.7、config-server

待补充

## 3.8、mapreduce

待补充

## 3.9、mapreduce 计算结果入库

一段独立的客户端程序，负责将 hdfs 上的结果数据发送给 dbserver 进行入库操作

待补充