**Php直接从后台程序交互的协议部分：**

**设置cdn最大速度的协议号为6001**

**设置cdn速率分段的协议号为6002**

**TCP协议头**

**typedef struct protocol {**

**uint32\_t len;//协议头及协议体数据的总字节数**

**uint32\_t seq;//置0**

**uint16\_t cmd;//6001或6002**

**uint32\_t ret;//置0**

**uint32\_t id; //置0**

**} \_\_attribute\_\_((packed)) protocol\_t;**

**协议体：**

**设置cdn最大速度**

**uint32\_t max\_speed;//4字节**

**设置速率分段**

**uint32\_t speed\_dis[4];//16字节**

**基本上和之前请求全国地图的协议一致。不过由于之前的协议只有协议头，没有协议体，所以len要适当修改。之前的len是18，代表协议头的18个字节长度。**

**如果是设置cdn的最大速度 那len 现在要变成22(18+4) 而且在协议头后面附带上1个4字节的max\_speed值,cmd为6001**

**如果是设置cdn的速率分段 那len 现在要变成34(18+16) 而且在协议头后面附带上4个4字节的速率分段值,cmd为6002**

**协议头的seq ret id都置0.**

**无论设置成功与否，后台程序都会返回一个协议头。如果成功 ret=0 如果失败 ret为其他值。**

**IDC全国地图协议不变。**

**CDN全国地图协议部分: 请求数据的协议和返回数据的协议头都变，但是返回数据的协议体增加了一个字段，用来表示客户端数量分布。如下：**

typedef struct p\_cache\_cdn\_area{

uint32\_t province\_code;

uint32\_t avg\_speed;

**uint32\_t speed\_distribution[6];**

}\_\_attribute\_\_((packed)) proto\_cache\_cdn\_area\_t;

typedef struct p\_cache\_cdn\_node{

uint32\_t node\_value;

uint32\_t province\_code;

uint32\_t city\_code;

uint32\_t avg\_speed;

**uint32\_t speed\_distribution[6];**

}\_\_attribute\_\_((packed)) proto\_cache\_cdn\_node\_t;

**第1个表示错误的客户端数量**

**2表示[0-A) 客户端数量**

**3表示[A-B) 客户端数量**

**4表示[B-C) 客户端数量**

**5表示[C-D) 客户端数量**

**6表示[D-max\_cdn\_speed] 客户端数量**

**其中 A B C D为协议6002设置的数值。**

**CDN节点的来访地区曲线图部分：**

**数据库名**

**cdn\_rate\_speed\_dis\_01 小花仙 显示节点的客户端速率分布**

**cdn\_rate\_detail\_01 小花仙 画节点的来访地区曲线图**

**cdn\_rate\_speed\_dis\_03 摩尔**

**cdn\_rate\_detail\_03 摩尔**

**cdn\_rate\_speed\_dis的表名**

**按cdn节点分**

**t\_cdn\_speed\_dis\_cdnipxx\_lv0**

**t\_cdn\_speed\_dis\_cdnipxx\_lv1**

**t\_cdn\_speed\_dis\_cdnipxx\_lv2**

**t\_cdn\_speed\_dis\_cdnipxx\_lv3**

**字段 ip node seq provid cityid isp speed\_tag count**

**主键(ip node seq provid cityid isp speed\_tag)**

**xx是节点号模100的值 cdnip与节点号的对应关系依然是 node = cdnip>>8 或者node = cdnip/256**

**按省分**

**t\_cdn\_speed\_dis\_provxxxxxx\_lv0**

**t\_cdn\_speed\_dis\_provxxxxxx\_lv1**

**t\_cdn\_speed\_dis\_provxxxxxx\_lv2**

**t\_cdn\_speed\_dis\_provxxxxxx\_lv3**

**字段 ip node seq cityid isp speed\_tag value count**

**主键(ip node seq cityid isp speed\_tag)**

**xxxxxx是省编号**

**speed\_tag表示速率属于哪个速率区间。每10K一个区间**

**如果speed\_tag=10 则表示这条数据表示的是速度为100-110的客户端数 speed\_tag = -1表示错误的客户端数。具体的数目由count表示。**

**Cdn\_rate\_detail的表名：**

**按cdn节点分**

**t\_cdnrate\_cdnipxx\_lv0**

**t\_cdnrate\_cdnipxx\_lv1**

**t\_cdnrate\_cdnipxx\_lv2**

**t\_cdnrate\_cdnipxx\_lv3**

**字段 ip node sed provid cityid comp value count**

**主键(seq, ip, provid, cityid, comp)**

**按省分**

**t\_cdnrate\_provxxxxxx\_lv0;**

**t\_cdnrate\_provxxxxxx\_lv1;**

**t\_cdnrate\_provxxxxxx\_lv2;**

**t\_cdnrate\_provxxxxxx\_lv3;**

**xxxxxx表示省编号。**

**字段 cityid comp ip seq value count**

**主键 (seq, ip, cityid, comp)**

**比如对于小花仙的一个节点node1 要查看来访地区 prov1的电信曲线图及其客户端分布。**

**曲线图数据在cdn\_rate\_detail\_01中**

**xx = node1%100 假如为1 且假如你查的是小时粒度的数据 则在lv0中查(和以前的规则一样)**

**在t\_cdnrate\_cdnip01\_lv0中查找node=node1 provid=prov1 isp=电信的数据 按seq 做group 得出来的seq为横坐标**

**sum(value)/sun(count)为纵坐标**

**某一时间点seq1的客户端分布在数据库cdn\_rate\_speed\_dis中的t\_cdn\_speed\_dis\_cdnip01\_lv0 中查找 node=node1 provid =prov1 isp=电信 seq=seq1 的数据。如果查50K-100K的数据的客户端量再加上speed\_tag 大于等于5且小于10。sum(count)就是客户端量。**