

xbull功能说明

名词解释：

端点：由一个IP地址以及一个TCP端口组成，该IP地址与TCP端口对应一个地磅的数据串口；地磅的数据串口通过一个TCP转换设备转换为TCP连接，安装时需要在TCP转换设备中配置好，使得服务方式为 TCP 服务器模式；端点对应一个数据协议分别为1，2，3号协议，协议中通常也规定了数据的单位为吨或者公斤，配置中需要指明该端点支持的协议以及数据单位；

xbull 作为端点数据采集器，通过TCP协议连接到该端点，并从端点读取数据，根据配置的数据协议解析出重量信息，如果数据单位配置为公斤，则转换为吨；

煤矿：xbull运行于一台带CPU的Linux机器中，该机器需要配置一个参数，即该机器的归属，称之为煤矿编码，该编码代表一个煤矿；（归属单位）

COLLIERYID

地磅：端点实际上连接到一台地磅，每台地磅在同一个煤矿中，需要有唯一的编码，规定为3个数字编码，其中编码 000 保留为应用本身。应用本身与一个本地运行的MYSQL数据库相关联，因此在两台不同的机器中 配置的煤矿编码相同时，两台机器中的应用编码都是000，但两台机器所连接的地磅则需要各不相同的编码。

协议：

协议1：

所有数据均为ASCII码，每帧数据共有8字节组成（包括包括小数点），数据传送先低位后高位，每帧数据间有一组分隔符“=”，发送数据为当前显示称重值，如当前显示称重重量为188.5，连续发送5.88100=5.88100=.....。如当前显示称重重量为-1885，连续发送.58810=-.58810=-.....。

协议2：

所有数据均为ASCII码，每帧数据共有9字节组成（包括小数点），数据传送先低位后高位，每帧数据间有一组分隔符“=”，发送数据为当前显示称重值，如当前显示重量为188.5，连续发送5.88100=5.881000=.....。如当前显示正中重量为-1885，连续发送.588100=-.588100=-.....。

协议3：

所传送的数据为仪表显示的当前重量。每帧数据由12组数据组成。格式如下表所示。
异或=2 xor 3 xor 4 xor 5 xor 6 xor 7 xor 8 xor 9。

第 X 字节	内容	注 解		举 例 (发送 +20.00)	
		内容	代码	内容	十六进制代码
1	开始	(XON)	02	XON	02
2	+或-	符号位	2B/2D	+	2B
3	称量数据	最高位	30~39	0	30
4			30~39	0	30
5			30~39	2	32
6			30~39	0	30
7			30~39	0	30
8		最低位	30~39	0	30
9	小数点位数	从右到左 (0~4)	30~34	2	32
10	异或校验	高四位		异或校验 =0x1B	01
11		低四位			0b
12	结束	XOFF	03	XOFF	03

应用程序环境要求

xbull 可执行程序运行时，会在当前运行目录生成 xbull.log xbullErr.log 以及 xbull.json，因此需要在可写目录作为当前目录运行xbull程序

xbull.log 当命令行指定 -log 参数时,xbull将生成详细运行日志

xbullErr.log 当xbull 遇到重大错误时，会在xbullErr.log中生成日志，如协议配置错误，以及数据插入发生错误时

xbull.json, xbull启动时，从数据库读出config 表以及endpoints 表的内容生成xbull.json文件，可以查看配置是否正确

```

192:xbull kennshi$ ./xbull -h
-dsn string
    DataSourceName Of Mysql (default "root:13811237916sS@tcp(localhost:3306)/bridge")
-h      print this help
-log
    enable log
-testDB
    test mysql
192:xbull kennshi$

```

当需要详细运行日志时，可指定 -log参数

当需要测试dsn是否工作时，可指定 -testDB参数

xbull 将使用 -dsn指定的数据源字符串连接数据库，如果连接成功，写入一条Data数据，以及一条Event数据，并从数据库读取config表和endpoints表生成json格式的配置数据，打印到标准输出。

正常运行时，需要从命令行指定 -dsn参数，注意密码不能包含特殊字符

应用程序采用单一进程模式，在应用程序的环境中应该有进程监视程序，可采用 **supervisor** 或者 **supervisord**，或者其他进程监视程序

功能要求

一个应用程序，可以连接若干端点，可连接的数量取决于机器性能，主频，网口吞吐量以及内存大小

- 原需求：数据要稳定值；
- 新需求：数据变化就需要(数据需满足最小值以及存在最短时长)，包括0值
- 新需求：数据表中主键为**UUID varchar(32)**

配置表

- (tableName: config)
- 本煤矿的编号 (9位数字)
- 应用程序心跳时长 (单位为秒)
- 端点的数量
- 端点数据超时时长 (单位为秒)
- 端点数据持续多长时间才有效 (单位为秒)
- 端点数据大于多少吨才有效 (单位为吨，浮点数)

```
create table config
(
    item      varchar(32) null,
    value     varchar(16) null,
    itemtype  char        null,
    comment   varchar(128) null
);
```

	item	value	itemtype	comment
1	id_colliery	140000000	S	本煤矿的编码
2	interval_heartbeat_app	600	N	应用程序心跳间隔(秒)
3	timeout_endpoint_data	600	N	端点数据超时时长(秒)
4	duration_endpoint_data	10	N	端点数据持续多少时间才有效(秒)
5	min_endpoint_data	1.5	N	端点数据大于多少吨才有效(吨)
6	num_endpoint	3	N	端点的数量
7	interval_reconnect	60	N	短线重连间隔(秒)
8	event_epdata_change	false	T	是否生成端点数据改变事件
9	save_epdata_zero	false	T	是否存储值为零的数据

itemtype的值只能是大写字母 S,N,T 三者之一， S表示该项为字符串， N表示该项为数值， T表示该项为布尔值

(tableName:endpoints)

端点端口的配置： IP地址， TCP端口， 端点吐出数据的单位（吨或者公斤）端点的的协议（1， 2， 3）， 地磅的编号（三位数字）

```
create table endpoints
(
    ipaddr varchar(15) null,
    tcpport int null,
    unit tinyint null
    comment '0,单位为吨; 1, 单位为公斤',
    protnb tinyint null
    comment '协议编号, 只能是1, 2, 3',
    epid char(3) null
);
```

	ipaddr	tcpport	unit	protnb	epid
1	127.0.0.1	6000	1	1	001
2	127.0.0.1	6001	1	2	002
3	127.0.0.1	6002	1	3	003

单位为公斤时，xbull将收到的值除1000转换为吨。

编码表

事件编码表 见数据库对应表格

```
create table eventlist
(
    eventid char(2) null,
    eventtext varchar(255) null
);
```

	eventid	eventtext
1	A0	应用程序启动
2	A1	应用程序心跳
3	E0	端点连接成功
4	E1	端点连接中
5	E2	端点连接断开
6	E3	端点连接心跳(不使用)
7	E4	端点数据超时
8	E5	端点数据变动

编码表和配置表为基础数据，安装时需要录入数据，编码表xbull程序并不使用，xbull程序中直接使用编码值

xbull程序运行时，产生事件，并插入事件表，其中E5事件（端点数据变动事件）可能会比较多，所以在Config表中有一项配置，可决定是否产生该项事件

事件表

```

create table eventlog
(
    recid          int auto_increment
    | primary key,
    collieryid    char(9)          null,
    bridgeid      char(3)          null,
    eventtime     timestamp        null,
    eventid       char(2)          null,
    eventinfo     varchar(255) null
);

```

其中A0 事件的info为程序版本号

E0及E1事件的info为连接的IP及端口 真实的地磅数据会连续不断吐数据，所以E4数据超时事件不会产生，有可能串口与 TCP转换设备之间出现问题，此时表现为TCP连接存在，但xbull收不到任何数据，就会产生E4事件。并且会连续产生，直到收到数据为止。 同样，E1事件也有可能连续产生。

	recid	collieryid	bridgeid	eventtime	eventid	eventinfo
1	5286	140000000	000	2019-11-02 02:02:25	A0	1.0.10
2	5287	140000000	001	2019-11-02 02:02:25	E1	(127.0.0.1:6000)
3	5292	140000000	001	2019-11-02 02:02:25	E0	(127.0.0.1:6000)
4	5293	140000000	000	2019-11-02 02:13:16	A1	1.0.10
5	5294	140000000	001	2019-11-02 02:13:37	E4	
6	5299	140000000	001	2019-11-02 02:40:00	E1	(127.0.0.1:6000)
7	5301	140000000	001	2019-11-02 02:40:00	E0	(127.0.0.1:6000)
8	5409	140000000	001	2019-11-02 02:45:19	E2	(127.0.0.1:6000)
9	5410	140000000	001	2019-11-02 02:45:19	E1	(127.0.0.1:6000)
10	5411	140000000	001	2019-11-02 02:46:19	E1	(127.0.0.1:6000)

E5事件的info为旧值 收到旧值的次数 第一次收到旧址值的时间戳 旧值存在秒数
新值

74	5359	140000000	001	2019-11-02 02:40:19	E5	OLD:48,940 3 2019-11-02 02:40:18.719 0 New:48,960
75	5360	140000000	001	2019-11-02 02:40:19	E5	OLD:48,960 1 2019-11-02 02:40:19.031 0 New:48,980
76	5361	140000000	001	2019-11-02 02:40:20	E5	OLD:48,980 4 2019-11-02 02:40:19.136 0 New:49,000
77	5362	140000000	001	2019-11-02 02:40:22	E5	OLD:49,000 20 2019-11-02 02:40:19.550 2 New:48,980
78	5363	140000000	001	2019-11-02 02:40:23	E5	OLD:48,980 9 2019-11-02 02:40:21.617 1 New:49,000
79	5364	140000000	001	2019-11-02 02:40:23	E5	OLD:49,000 1 2019-11-02 02:40:22.550 0 New:49,020
80	5365	140000000	001	2019-11-02 02:40:24	E5	OLD:49,020 14 2019-11-02 02:40:22.653 1 New:49,000
81	5366	140000000	001	2019-11-02 02:40:25	E5	OLD:49,000 5 2019-11-02 02:40:24.108 1 New:48,940
82	5367	140000000	001	2019-11-02 02:40:37	E5	OLD:48,940 115 2019-11-02 02:40:24.625 12 New:48,960
83	5368	140000000	001	2019-11-02 02:40:37	E5	OLD:48,960 1 2019-11-02 02:40:36.532 0 New:49,020
84	5369	140000000	001	2019-11-02 02:40:37	E5	OLD:49,020 7 2019-11-02 02:40:36.636 1 New:49,000
85	5370	140000000	001	2019-11-02 02:40:38	E5	OLD:49,000 6 2019-11-02 02:40:37.366 1 New:48,980
86	5371	140000000	001	2019-11-02 02:40:40	E5	OLD:48,980 21 2019-11-02 02:40:37.992 2 New:49,000
87	5372	140000000	001	2019-11-02 02:40:42	E5	OLD:49,000 21 2019-11-02 02:40:40.168 2 New:49,020

数据过程表

```
create table weightproc
(
    RecID          varchar(32)  not null
    primary key,
    WeightTime     datetime      null,
    WeightValue    decimal(8, 2) null,
    VehNum         varchar(10)   null,
    CollieryID     char(9)       null,
    BridgeID       char(3)       null
);
```

RecID	WeightTime	WeightValue	VehNum	CollieryID	BridgeID
1 e4d551c2fcd811e99f3b0242ac110002	2019-11-02 02:53:29	0.00		140000000	003
2 f500b387fcd811e99f3b0242ac110002	2019-11-02 02:53:56	48.94		140000000	003
3 fd092692fcd811e99f3b0242ac110002	2019-11-02 02:54:10	0.00		140000000	003

该表中数据的插入取决于config表中两个项
min_endpoint_data 值必须大于等于该值才会被插入表中
duration_endpoint_data 值必须连续相等，且存在时长大于等于该值
在同一辆车产生的地磅数据中，有可能多个数据满足上述条件
weightValue 为 零值的记录是否产生，取决于config表中 save_epdata_zero的
值。

数据表

```
create table weightdata
(
    RecId      varchar(32)    not null
    | primary key,
    CollieryId char(9)        not null,
    BridgeId   char(3)        not null,
    VehiNum    varchar(10)    not null,
    BeginTime  datetime       not null,
    EndTime    datetime       not null,
    WeightValue decimal(8, 2) not null
);
```

该表记录 由零变为非零（上磅时间），再次变为零的时候（下磅时间），持续时间最长的重量数据。

备注：xbird 为模拟程序

```
192:xbird kennshi$ ./xbird_macos -h
-comport string
    name of comport when comtype=='com' (default "COM2")
-comtype string
    comm type, 'com' or 'tcp' (default "tcp")
-h print this help
-listcomport
    list com port
-protocol int
    protocol Number: 1,2,3 (default 1)
-tcpport string
    tcp port number when comtype=='tcp' (default "6000")
-test
    execute testcase..
```

需要在命令行指定 侦听端口和协议类型 输入 data+回车键，程序连续送出一组真实的地磅数据，单位为公斤

2019-12-18 新增配置与连通性测试管理

将 start.sh 与 xbull 拷贝到目标机器的可写目录

命令行执行 `nohup ./start.sh >/dev/null 2>&1 &` 启动 xbull

通过浏览器 `http://目标机器 IP 地址 :8864` 打开配置与连通性测试管理如下图：

XBull 配置与连通性测试管理

煤矿编号:	<input type="text" value="140000000"/>
Dsn:	<input type="text" value="root:13811237916sS@tcp(localhost:3306)/bridge"/>
心跳消息间隔(秒):	<input type="text" value="600"/>
网络重连时间间隔(秒):	<input type="text" value="60"/>
数据超时时间间隔(秒):	<input type="text" value="600"/>
数据持续时间(秒):	<input type="text" value="10"/>
数据最小值(吨):	<input type="text" value="1.5"/>
设备数量:	<input type="text" value="4"/>
<input type="button" value="确认"/>	
数据值变化事件是否写入数据库: <input checked="" type="checkbox"/>	
零值数据是否写入数据库: <input checked="" type="checkbox"/>	

设备列表

设备编号	设备IP地址	TCP端口	数据单位	协议编号
001	192.168.111.200	6001	吨	3号协议
002	192.168.111.201	6001	吨	3号协议
003	127.0.0.1	6004	公斤	3号协议
004	127.0.0.1	6005	公斤	3号协议

提交

连通性测试

如果启动时，没有配置数据库参数，可通过这个界面配置数据库参数，提交之后，重新打开该界面，配置其他参数。

设备数量修改之后，需要先点击“确认”使得设备列表更新。修改完毕后，点击提交按钮，xbull 将重启，并使用新参数工作。。

重新刷新该页面，可以点击 “连通性测试” 按钮，获得所配置的设备列表的连通性

XBull 配置与连通性测试管理

煤矿编号:

140000000

Dsn:

root:13811237916sS@tcp(localhost:3306)/bridge

心跳消息间隔(秒):

600

网络重连时间间隔(秒):

60

数据超时时间间隔(秒):

600

数据持续时间(秒):

10

数据最小值(吨):

1.5

设备数量:

4

确认

数据值变化事件是否写入数据库:

☒

零值数据是否写入数据库:

☒

设备列表

设备编号	设备IP地址	TCP端口	数据单位	协议编号	
001	192.168.111.200	6001	吨	3号协议	
002	192.168.111.201	6001	吨	3号协议	
003	127.0.0.1	6004	公斤	3号协议	
004	127.0.0.1	6005	公斤	3号协议	

提交

连通性测试

如上图，设备列表的最后一列，显示连通性。。 黄色表示，正在连接设备，还没有返回结果， 绿色表示正常连通， 红色表示不能连接。。

用户登陆功能：

创建一张表

```
create table users
(
    name varchar(16) null,
    salt  varchar(16) null,
    passwd varchar(32) null
);
```

插入一条数据， salt 为插入表时设置的随机字符串， 密码字段的值为

md5(concat(salt, passwd))

例如：

```
insert into users (name,salt,passwd) values ("admin","FvscE",
md5(concat("FvscE","13910580009")));
```

判断是否合法，使用下列语句：

```
"select passwd = md5(concat(salt,'13910580009')) as loginok from users where
name= 'admin';"
```

您需要登录！

用户名:

密码:

输入用户名， 密码之后， 如果正确， 出现配置界面：

采集软件配置与连通性测试

煤矿编号: 140000000

Dsn: root:13811237916s5@tcp(localhost:3306)/bridge

心跳消息间隔(秒): 600

网络重连时间间隔(秒): 60

数据超时时间(秒): 600

数据持续时间(秒): 10

数据最小值(吨): 1.6

设备数量: 3

数据值变化事件是否写入数据库: ☒

零值数据是否写入数据库: ☒

设备列表

设备编号	设备IP地址	TCP端口	数据单位	协议编号
001	192.168.111.200	6001	吨	3号协议
002	192.168.111.201	6001	公斤	3号协议
003	127.0.0.1	6004	公斤	3号协议

比较之前的界面， 多了一个退出按钮； 点击退出按钮， 会退回到登录页面！