# RealGO v2.0.0功能说明。

V2.0.0版本增加了两个重点功能，1、读取分钟数据ReadMin。2、集群(cluster)功能。故做一个较大的升级。

**读取分钟数据ReadMin**

之前RealGo的存储颗粒度为秒，当采样周期较大（比如分钟）时，数据库仍然需要读取大量的秒级数据然后再慢采样为分钟，读取效率相对较低。体现在性能表现上，客户端读取一天左右采样间隔大于一分钟的数据性能较低。读取信噪比为1：60。

新版本重新设计了对于类似分钟数据这类统计量的存储办法，数据库运行时自动加载进行以分钟为单位的统计信息的加速存储引擎，分钟统计量的存储位置为datas文件夹的statmin文件夹中。以后可以扩展对各种各样类型的统计量（比如小时阀门操作次数）进行加速存储。现阶段statmin（分钟统计）对于采样间隔为分钟的数据读取信噪比为1：1，且可对程序意外死亡后重启的日志进行回滚，保证与原始秒级表的*数据一致性*。

RealGoDBDriver.dll程序中增加ReadDataMin接口对分钟级数据进行读取。

此功能已进行了上百万次的随机地址、随机开始时间、随机长度自动化测试。

**集群(cluster)功能**

集群功能添加之前需要对RealGO程序进行较大改动。RealGO v2.0之前版本的数据时间戳为RealGo程序内部自己根据本机时间产生。这样当程序运行在不同电脑上时，时间存在不一致的现象，无法实现集群数据访问。针对以上问题，修改了内部结构，当前RealGo的数据时间戳需要数据采集客户端程序在写入数据库时附加。但客户端程序需要保证每次写入时间戳必须大于之前的时间戳，否则不能有效写入。

新的RealGoDBDriver.dll程序使用集群方法如下，首先将libs文件夹中的HPSocket.Net.dll和HPSocket4C.dll拷贝入客户端当前目录中，然后引用RealGoDBDriver.dll。

示例代码如下：

using RealGoDBDriver;

//

//这里定义cluster集群实例，同时传入一：本机（127.0.0.1）位与3100端口的数据库连接信息，

//3100端口的地址范围为0, 100。二：本机（127.0.0.1）位与3101端口的数据库连接信息，

//3101端口的地址范围为101, 200。

//RealGoDBCluster在使用前，需要先在3100和3101端口启动两个RealGo进程,在RealGO程序界面//中输入start -p 3100和start –p 3101

// RealGoDBCluster内部自动连接3100端口服务器，并自动将0-100号地址分配到3100数据库进//程中（3100数据库进程地址长度为101），将101-200号地址分配到3101数据库进程中（3101

//数据库进程地址长度为100）

RealGoDBCluster realgo\_cluster = new RealGoDBCluster(new RealGoDBClientDescriptor("127.0.0.1", 3100, 0, 100), new RealGoDBClientDescriptor("127.0.0.1", 3101, 101, 200));

// realgo\_cluster.Connect进行集群连接

bool isconnect = realgo\_cluster.Connect();

//之后可以调用realgo\_cluster.Enable查看realgo\_cluster是否准备就绪。

//写入方法

DateTime l\_dt = new DateTime(2022, 4, 16, 11, 0, 0);

ushort[] addrs = new ushort[10] { 1, 2, 3, 4, 5, 110, 111, 112, 113, 114 };

float[] vals = new float[10] { 1f, 5f, 10f, 15f, 20f, 25f, 30f, 35f, 40f, 45f };

// realgo\_cluster自动将数据写入两个数据库进程中

RealGoDBResult ret0 =realgo\_cluster.WriteDataWithTime(addrs, vals, l\_dt);

// realgo\_cluster自动将数据从两个数据库进程中读取秒级数据，合并后传给最终客户

//注意ReadData方法返回值RealGoDBResult中Results为RealGoDBClusterReadResObj类型

//RealGoDBClusterReadResObj中ResultValues属性为请求返回值。ResultModes为每个index

//的返回类型OK、Failed、Timeout等。

RealGoDBResult ret1 = realgo\_cluster.ReadData(new ushort[7] { 1, 2, 3, 4,111,112,113 }, new DateTime(2022, 4, 16, 11, 1, 0), 40);

// realgo\_cluster自动将数据从两个数据库进程中读取分钟级数据，合并后传给最终客户

RealGoDBResult ret2 = realgo\_cluster.ReadDataMin(new ushort[7] { 1, 2, 3, 4,111,112,113 }, new DateTime(2022, 4, 16, 11, 1, 0), 10);

注意：目前对DataView客户端的升级改造尚未完成。

read，请求格式

read|body长度|8位随机码|4位开始时间戳|4位duration|读取地址ushort数组）

如read(new ushort[2]{1,2},1000000,128)

那么请求命令格式为

1-4，四字节，值为"read"(asc编码）。

5-8，四字节，后续内容的长度（int类型）。

9-16，八字节，随机码标识（ulong类型）。

17-20，四字节，起始时间减去2017年1月1日0时0分0秒的总秒数。

21-24，四字节，取数据长度（int类型），本例为128.

25-28，四字节，由两个ushort数据组成，为待取的地址数组。

read，返回格式

read|body长度|8位随机码|1为返回码|结果

如read(new ushort[2]{1,2},1000000,128)，他的返回格式为

1-4，四字节，值为"read"(asc编码）。

5-8，四字节，后续内容的长度（int类型）。

9-16，八字节，随机码标识（ulong类型）。需要检查与请求的随机码标识一致，否则丢弃。

17，一字节，返回码，含义如下

Failed=0,

OK =1,

Timeout=100,

DateTimeError=101,

ArgumentOutOfRange=102,

InvalidOperation=103,

Busy = 200,

Null=201,

NotFound=255

18-1041，共1024字节，前512字节为第一个地址的128个float返回值，后512字节为第二个地址的128个float返回值，以此类推。

write，请求格式

writ|body长度|8位随机码|4位版本号(未启用仅仅起到占位作用，放0000，4个字节即可)|4位时间戳|写入数值数组（两位地址(ushort)+四位数据(float)）

请求命令格式为

1-4，四字节，值为"writ"(asc编码）。

5-8，四字节，后续内容的长度（int类型）。

9-16，八字节，随机码标识（ulong类型）。

17-20，四字节，版本号（未启用仅仅起到占位作用，放0000，4个字节即可）。

21-24，四字节，起始时间减去2017年1月1日0时0分0秒的总秒数。

25-后续，由多组地址-数据对组成，每组长度为6，格式为ushort（地址）+float（地址对应值）。

write，返回格式

1-4，四字节，值为"writ"(asc编码）。

5-8，四字节，后续内容的长度（int类型）。

9-16，八字节，随机码标识（ulong类型）。需要检查与请求的随机码标识一致，否则丢弃。

17，一字节，返回码，含义如下

Failed=0,

OK =1,

Timeout=100,

DateTimeError=101,

ArgumentOutOfRange=102,

Busy = 200,

Null=201,

NotFound=255