

高支模走查

提取软件

1. ExtractService.cs **ExtractData()** 120 *优化*

函数体太长，建议按内容分出，“配置”、“查询”、“上报”三个函数

2. ExtractService.cs **ExtractData()** 150 *建议改善*

配置读取可拿到外面定时刷新，刷新间隔可以小时级别粒度

3. ExtractService.cs **ExtractData()** 141/184 *需要改善*

是否可以使用ID作为查询游标。毕竟在 A0时间列上没有索引，查询分析还是要遍历表内所有记录，影响效率。

```
string queryStr = "select * from " + TableName + " where A0>" + s_beginTime  
+ "'and A0<" + endTime.ToString("yyyy-MM-dd HH:mm:ss") + "'";
```

4. ExtractService.cs **ExtractData()** 198 *优化*

可做常量

```
System.DateTime startTime = TimeZone.CurrentTimeZone.ToLocalTime(new  
System.DateTime(1970, 1, 1)); // 当地时区
```

5. ExtractService.cs **ExtractData()** 184 *需要改善*

建议增加SQL LIMIT子句

```
string queryStr = "select * from " + TableName + " where A0>" + s_beginTime  
+ "'and A0<" + endTime.ToString("yyyy-MM-dd HH:mm:ss") + "'";
```

通过LIMIT+循环处理，将长查询拆分为多个短查询，在查询时间内，释放一定的CPU和内存资源。

6. ExtractService.cs **ExtractData()** 350 *优化*

放到 final

```
dbConnection.Close();
```

7. ExtractService.cs **ExtractData()** 307 *优化*

根据以上逻辑，没有限制一次取的数据量，这里可能导致包体长度过大，在处理json化和http传输时对内存可能有压力

```
result allData = new result() { allNodeDatas = senorDataList };  
try  
{  
    string jsonData = Converter.Serialize(allData); //序列化数据
```

采集软件

BOOT 是启动程序（看门狗程序）

根据tt.txt中的boottime来判断，来监控采集软件是否启动。定时20秒检查