УДК 004.652

WAYS FOR IMPROVING METHODS OF DATA STORING IN MONITORING SYSTEMS

H. Hussein, L.I. Suchkova, M.A. Yakunin
Minia university,
Egypt
Altai State Technical University. II Polzunov
Altai State University
Barnaul, Russia

The article is devoted to the development of an approach that allows no loss of accuracy to reduce the amount of data stored on a server while reducing the time to access the database.

Keywords: optimization of field types, databases, SCADA-systems.

The main tool environment for solving problems of temperature and environmental monitoring, as well as various systems of commercial and operational control and metering of energy resources (such as AMR and ASKUT) today are SCADA - system. On the one hand, the application based SCADA - systems can quickly deploy a large software and hardware, but on the other hand, because of its versatility it is inherent in a number of significant deficiencies, of which the most important is the non-optimal way to store the data. The fact that the basic requirement for monitoring systems and, in particular, environmental monitoring and energy accounting is the ability to play back the recorded data over long time periods, up to several years. The dynamics of the registered processes in individual cases suggests that the sample period does not exceed 5-30 seconds. Given also that the number of different types of data storage can be tens or even hundreds, database size quickly reaches a value at which time data sampling begins to significantly slow down the operator of such a system.

The aim of this work is to develop an approach that allows no loss of accuracy to minimize the amount of data stored on a server while reducing the time to access the database.

The basis of this approach is the principle of optimization of the type of fields used for recording your own samples, and writing samples to meet these time points.

First of all, for such optimization was proposed to use for the storage of regularly updated data fields only fixed-length integer unsigned (unsigned). As the real samples can be positive or negative, and also have a fractional part, it is proposed to carry out before writing their normalization by the formula y = k * x + y0, where x - observed data, k - a factor, and y0 - offset. Since many databases provide a rich selection of fields of a type other than the number of bytes, you can always find this type of field in which at its minimum size provides a record of all controlled variable range of stored values. So, for example, to record temperatures, removed from digital sensors such DS18S20, lying in the range from -55 to +125 ° C with a resolution of up to 0.125 C °, enough two bytes instead of storing it in the form of a real number in the Size 4 bytes. And while you can store data up to 0.005 C °, if you select k = 200 and y0 = 1000, which will be used for recording temperature for improving the resolution averaging over several samples. And even for the storage of such quantities as the daily and monthly consumption of heat, water and electricity to most of the objects with sufficient commercial and records will be accurate enough 2 bytes.

Further reduce the size of the field may be due to the transition from the store values accumulated from the start (zero) to the values of the system, to accumulate in a relatively short intervals of time, just as it is done in a sigma-delta ADC and video compression systems. With this campaign can sometimes be sufficient to store the data in the fields of size up to 1 byte. However, given that the standard reporting forms for expenditure of energy have a three-column format, and involves finding the heat or energy as the difference between the meter readings for the current and previous period, it is advisable to add the main table of the auxiliary, which would have kept the initial values of meter readings at the time of Input monitoring system in operation.

The next step, let's go to at least halve the amount of data stored, is the transition from the field, intended to store the date and time (in ac-

cordance with ANSI / ISO SQL 92 is a field of type DATE, TIME, DATETIME, and TIMESTAMP [1]) is also to an unsigned integer fields. The practice of this approach showed that before such conversion desirable to convert the time to UTC - format, which eliminates the problem of time shifts due to irregular the clock for daylight savings time and administrative possible jet lag local time for a specific locality. Then to see the data for a particular period in a particular locality translation UTC - time to local time, you can either use stored in a separate database table data on time-shift and standard functions database or CGI - handler, or by writing your own conversion functions. For example, this time transformation UTC to local time can be done by using the following C # code.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;
public partial class UserDefinedFunctions
{
[Microsoft.SqlServer.Server.SqlFunction]
public static SqlDateTime ConvertToLocal-
Time (SqlDateTime utcTime)
{If (utcTime.IsNull)
return utcTime;
else
return new SqlDateTime (utc-
Time.Value.ToLocalTime ());
}};
```

At the heart of the code is the formula to convert the time:

UTC - (Bias/1440) = local time.

A constant value Bias, showing what time zone is on the local computer, available in the system registry

In determining the value of the time allotted for the reference field must follow the minimum required time period, the sample data and the required storage time of the data. In this case, the database will be stored values of the form ni = (ti-to) / Δt, where ti - sample times, to - boot time and Δt - sampling interval. Thus, in the following table 1 shows that, for most practical applications of this approach is sufficient to apply for temporary sampling field size from 1 to 3 bytes instead of 8 bytes obvodimyh for storing the date and time in most database systems. Since the lifetime of software without modifying the order of 3 - 7 years, in the table in bold fields that provide the most effective combination of data retention, the width of the field and the sampling interval data.

Other methods of data compression less universal, applicable only to a certain type of observed data. For example, check the temperature dependence in most cases is not important specific behavior of the curve, and it is important only general semantic image. As seen from the figure below, many thermal processes do require registration for accurate temporal resolution of the order of 10-30 s. However, in most cases, long-term storage of data is sufficient to record only the type of semantic image (pulsation, spontaneous emission, spikes, etc.), and the corresponding numerical parameters (period and amplitude of the fluctuations, the magnitude and duration of releases, etc.)

Table 1 – The dependence of the maximum storage time, years, on the sample period and the size of the field.

Count. Byte	1c	10s	1min	5min	10 min	30 min	1 hour	24 hours	1 mes.
1								0.7	21
2					1.3	4	7.5	180	
3	0.53	5.3	32	160	333				
4	136								

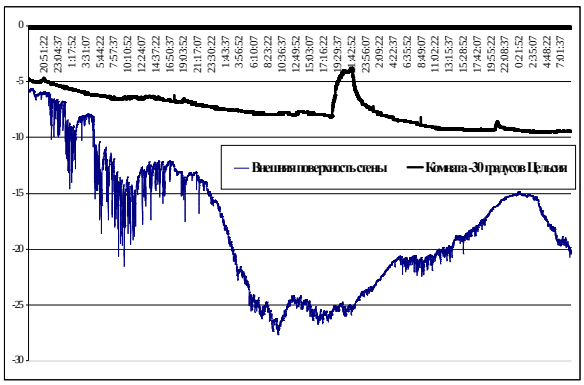

Figure 1 – The impact of wind gusts on the temperature dependence for street wall top graph (room temperature) for clarity shifted to -30 ° C

As an alternative to this approach can offer data compression by fixing on some time interval of maximum and minimum values of the controlled quantities, just as it is done in trading systems (known as "candlesticks"). Only in some cases, when the stationary curve appear transient changes are a specific type does not fit into the list of accepted semantic images (as shown for example in Figure 3) may require the fixation of each sample of the observed dependence.

You can use the so-called "magnifying glass of time" when stored log, recording the time of occurrence and duration of the non-specific sites and separately – a table with all samples controlled variable in this area.

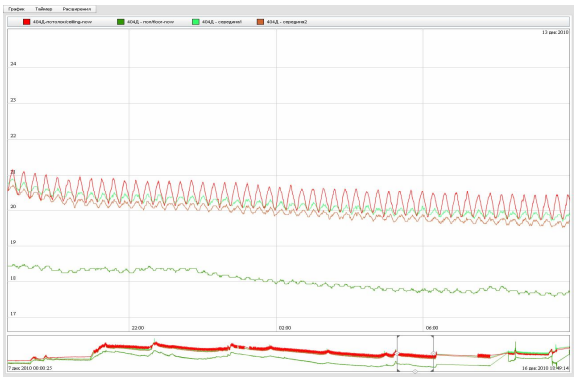

Figure 2 – Of temperature dependences under the influence of an electric heater

Clearly, to reduce the amount of stored data and apply the classical methods of data compression, basic, for example, on the approximation of the observed dependence of the linear or parabolic function. Certain effect can be and the rejection of record index files, because the data in the database tables are written sequentially and are selected (if not consider exceptions resulting data spooling after disturbance of the channels of communication). However, this approach depends on the database and evaluation of its use requires further study.

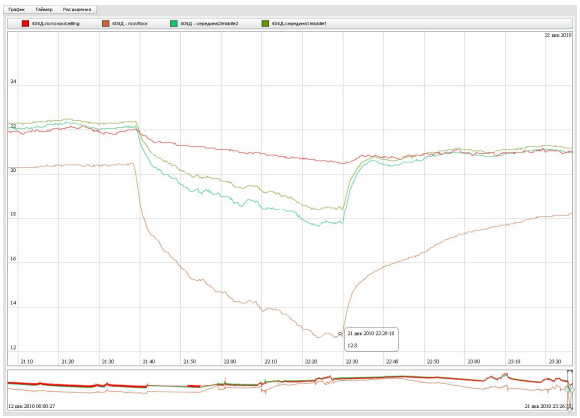

Figure 3 – The change of temperature curves at different vertical levels as a result of ventilation space.

LITERATURE

1. Kuznetsov MV Tutorial MySQL 5. / MV Kuznetsov, IV Simdyankin. - St.: BHV-Petersburg, 2007. - 560 p., Ill.

Hussein Hussein – graduate, tel.: (3852) 29-07-86, e-mail: ims_lab@agtu.secna.ru, Suchkova Larisa Innokent'evna – c.t.s., docent, Yakunin Maksim Alekseevich – graduate.