文章编号：0253-2697(2016)02-0266-07 DOI:10.7623/syxb201602014

一种抽油机示功图数据无损压缩存储方法

李金诺 龚仁彬 李 群 王从镔 姚 刚

（中国石油勘探开发研究院西北分院 甘肃兰州 730020）

摘要：抽油机示功图数据的存储是油田信息化重点关注的问题之一。若将抽油机示功图存储在实时数据库中，每口井每幅需要投入 500 个数据点进行存储，成本高昂；传统的抽油机示功图存储方法是将示功图数据存储在关系数据库中，此方法读取效率不高，不便于进行实时数据分析，且需要在油田生产作业区及作业区以上组织机构部署关系数据库服务器，代价依然昂贵。生产物联网示功图数据压缩方法，经过超过 2×10^4 幅示功图数据的计算、分析和验证，可以通过数据处理、精度确认、求取差值和数据存储 4 步计算将抽油机示功图数据有效压缩至原数据大小的 1/250，直接存储于实时数据库中，实现了降本增效的目的。该方法经过中国石油天然气集团公司相关油气田公司的试用，验证了其实用性，节约了投资成本。

关键词：抽油机；示功图；数据压缩；实时数据库；物联网

中图分类号：TE933 **文献标识码**：A

A lossless data compression and storage method of pump dynamometer

Li Jinnuo Gong Renbin Li Qun Wang Congbin Yao Gang

(*Northwest Branch, PetroChina Research Institute of Petroleum Exploration and Development, Gansu Lanzhou* 730020, *China*)

Abstract: Data storage of the pump dynamometer is one of key issues in oilfield informatization. If the pump dynamometer data are stored in real-time database, 500 data nodes should be used to store a piece of data for each well, thus leading to a high cost. In the use of traditional methods, dynamometer data are saved in relational database. However, this solution shows a lower reading efficiency, unbeneficial to real-time data analysis. Moreover, it is required to deploy database servers in oilfield production-operation zones and the organizations above the operation zones, and the cost is also higher. In application of the dynamometer data compression method for internet of things in production, through the calculation, analysis and validation on more than twenty thousand pieces of dynamometer data, pump dynamometer data can be effectively compressed to 1/250 of original data size by four steps, i.e., data processing, precision confirmation, difference solution and data storage. Such data can be saved in real-time database to reduce cost and improve effect. This solution has been adopted by several sub oil/gas field companies of CNPC to validate the practicability and save the investment cost.

Key words: pumping unit; dynamometer; data compression; real-time database; internet of things

引用：李金诺，龚仁彬，李群，王从镔，姚刚. 一种抽油机示功图数据无损压缩存储方法[J]. 石油学报，2016，37(2)：266-272.

Cite: Li Jinnuo, Gong Renbin, Li Qun, Wang Congbin, Yao Gang. A lossless data compression and storage method of pump dynamometer[J]. Acta Petrolei Sinica, 2016, 37(2): 266-272.

抽油机井是一种常见的采油井，中国内陆约 90%的油井都是抽油机井。在抽油机井上安装相应的采集设备对其工况进行实时监控，对其产量进行自动化计量是近年来工况诊断和产量计量常用的手段。对抽油机的生产数据进行实时采集形成抽油机示功图，通过对示功图数据的计算、处理和分析，实现对抽油机井远程监视、工况诊断和计量。计量数据也可进一步应用于产量预测[1,2]。由于该技术价格低廉，可实现对传统生产方式的改变，减少人工成本，提高生产效率，近年来在石油生产中得到了迅速的发展和广泛的使用[3]。

通常情况下，每半小时就需要采集一幅示功图数据，每幅示功图大约需要占用 500 个数据点，对一个油田来说，这个数据量是非常大的，会占用大量的存储空间和网络带宽，所以抽油机示功图的数据存储一直都是亟待解决的问题。由于实时数据库[4]是按数据点收费，如果采用实时数据库直接对功图数据进行存储，投资成本高昂。因此目前抽油机示功图数据存储主要使

基金项目：中国石油天然气集团公司油气生产物联网系统项目(A11)资助。

第一作者及通信作者：李金诺，女，1983 年 1 月生，2005 年获安徽大学计算机科学与技术专业学士学位，2012 年获北京大学软件工程专业硕士学位，现为中国石油勘探开发研究院西北分院工程师，主要从事油气生产物联网项目的数据库、功图研究、算法设计工作。Email: lijr_xb@petrochina.com.cn

用关系数据库，但这种存储方法无论从读取效率还是部署费用，依然存在着一些问题。所以本文提出了一种可以将抽油机示功图数据无损压缩后存储于实时数据库中的方法——示功图数据压缩方法，即 Dynachart Compression（Dynamometer card Compression），既节约了关系数据库的购置和维护费用，又使实时数据库的点数使用费用在可接受的范围之内。经过超过 2×10^4 幅示功图数据的检测和 3 家油气田公司的使用，确认了此方法的实用性，实现了节约资源、增加存储效率的目的。

1　现状分析

1.1　示功图原理和作用

用来表示悬点载荷与位移关系的示功图称为地面示功图或光杆示功图[5]，其是由载荷随位移的变化关系曲线所构成的封闭曲线图。在实际情况下，有多种因素影响示功图的形状，但每幅示功图都有其主要影响因素，所以示功图的形状反映着主要因素影响下的工况特征，利用示功图在不同因素影响下表现出来的特征可以进行产量计算和故障分析。

图 1 是地面示功图的示例图。其获取方法是通过在光杆上安装载荷位移传感器，测量光杆载荷变化及抽油机运行周期变化情况，经过信号转换即测出抽油机的地面示功图。

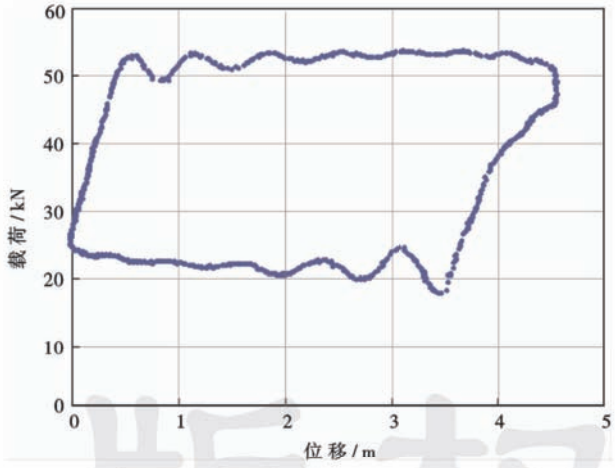

图 1　地面示功图

Fig. 1　Ground dynamometer

图 2 是井下泵功图，表示地层抽油泵处的位移和载荷之间的关系，泵功图可以用来做工况诊断。泵的典型示功图如图 3 所示。

泵的示功图共有 30 多种常见的工况，图 3 列出了几个比较典型的实例。

1.2　示功图传统存储方法

每幅示功图需要分别存储载荷和位移数据，每组数据约 200～300 个实数，共需要存储 500～600 个实数，每个实数通常要求保留到小数点后 6 位。

如果采用实时数据库对示功图数据进行直接存储，按照目前市场价格每个实时数据点 30 元/a 进行计算，每组示功图每年需要投入 500×30 = 1.5 万元进行购买存储该抽油机井示功图数据的实时数据库数据点，这个费用是非常高昂的。

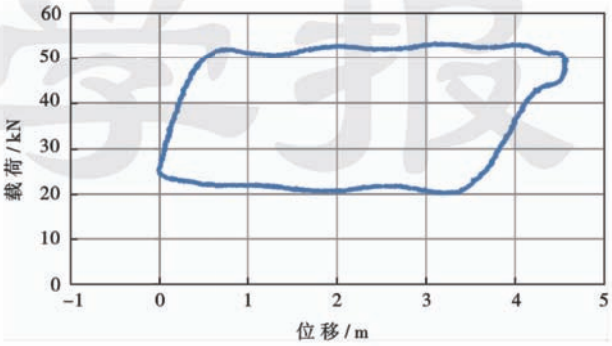

图 2　井下泵功图

Fig. 2　Pump dynamometer

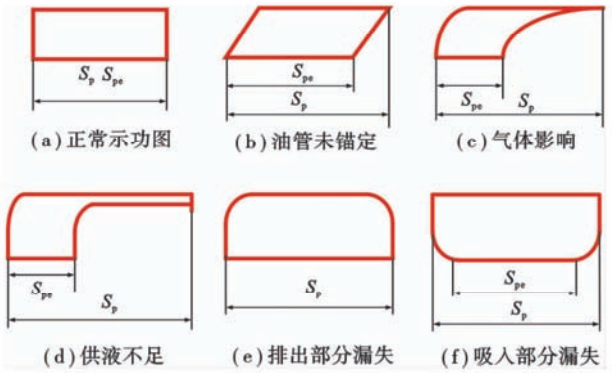

图 3　泵的典型示功图

Fig. 3　Common pump dynamometers

为了解决实时数据库的费用问题，目前通常的做法是采用关系数据库来对示功图数据进行存储。这种方法需要在每个作业区至少部署一台数据库服务器，并安装一套关系数据库软件，每套成本大约 10 万元。但该存储方法仍然存在 2 个问题：①利用关系数据库来进行实时数据的处理，其处理速度比实时数据库要慢很多，与 SCADA 软件的衔接需要进行数据转换，不利于实时动态分析；②若按照每个作业区 200 口井来计算，每个作业区需要投资 10 万元。中国石油天然气集团公司（简称中国石油）共有 30 多万口井、400 多个作业区，因此仅中国石油全部实施至少需要 5000 万元，该存储方式成本依然很高昂。

图 4 展示了关系数据库存储抽油机示功图数据的硬件部署。其中虚线框中的部分是专门为抽油机示功

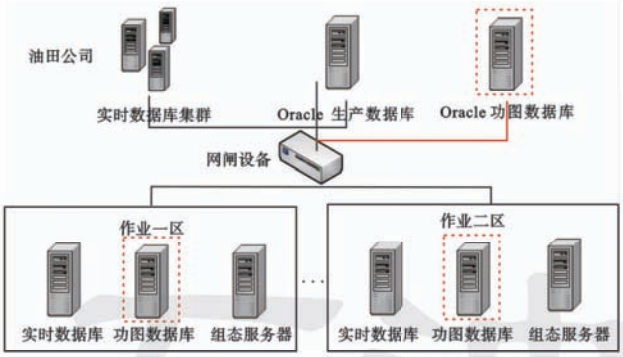

图 4 关系数据库存储示功图数据的硬件部署

Fig. 4 Hardware deployment diagram of relational database storage dynamometer data

图而部署的关系数据库。

综上所述，若能将示功图数据进行有效压缩，使用实时数据库进行存储，且将实时数据点的费用降在可接受的范围内，既可加快处理速度，也可大大节约投资。

2 示功图数据压缩算法

2.1 数据压缩算法

主流的数据压缩算法的核心是通过找到重复或有规律的数据来进行压缩，主要分为有损压缩和无损压缩两种[6-8]。

2.1.1 有损压缩

所谓有损压缩[9]是利用了人类对图像或声波中的某些频率成分不敏感的特性，允许压缩过程中损失一定的信息，因此不能完全恢复原始数据，即将次要的信息数据舍弃，牺牲一些质量来减少数据量，使压缩比提高。这种方法经常用于因特网尤其是流媒体以及电话领域。包括：PCM(脉冲编码调制有损压缩)、预测编码、变换编码插值和外推法、统计编码、矢量量化和子带编码等。

目前在实时数据库中主要使用的压缩算法是旋转门算法(SDT-Spinning Door Transformation)，也是一种有损压缩算法，这是一种比较快速的线性拟合算法，常常用于实时数据库中对数据进行压缩使存储容量有效地减少。

很显然，有损压缩方法并不能实现抽油机示功图数据的精确恢复，因此需要使用无损压缩算法，才能保证工况诊断的准确性。

2.1.2 无损数据压缩

无损数据压缩[10,11](Lossless Compression)是对压缩后的数据进行重构(或者叫做还原、解压缩)，重构后的数据与原来的数据完全相同，无损压缩数据经过压缩后信息不受损失，能完全恢复到压缩前。

无损压缩算法主要分为非熵编码和熵编码 2 种[12]：非熵编码原理是把原文的一段字母列被其他字母所取代；熵编码即编码过程中按熵原理不丢失任何信息的编码，信息熵为信源的平均信息量。

非熵编码：LZ77[13]与 LZ78[14,15]算法由 Abraham Lempel 和 Jacob Ziv 提出，这 2 个算法是大多数 LZ 算法变体(LZW[16]、LZSS 以及其他一些压缩算法)的基础。与最小冗余编码器或者行程长度编码器不同，这 2 个都是基于字典的编码器。LZ77 是“滑动窗”压缩算法，这个算法后来被证明等同于 LZ78 中首次出现的显式字典编码技术。

经过编码器和解码器都必须保存一定数量最近的数据，如最近 2 KB、4 KB 或者 32 KB 的数据。保存这些数据的结构叫滑动窗口，因为这个原理所以 LZ77 有时也被称作滑动窗口压缩。编码器需要保存这个数据查找匹配数据，解码器保存这个数据解释编码器所指代的匹配数据。但是，对于很少出现重复数字的 500 个示功图数据来说，很难找到匹配数据，因此用滑动窗口法进行数据压缩显然是不能达到压缩效果的。

熵编码法也是一种无损数据压缩技术，其特点是一段文字中的每个字母被一串不同长度的比特所代替。要使得所有的字母可以在压缩后互相区别需要保证每个字母被取代的比特数不能无限小，每个字母按照其出现的可能性所获得的最佳比特数取决于熵。熵编码的优点：① 编码表只需计算一次，因此编码速度高；② 除在解码时所需要的机率值外，结果肯定不比原文长。熵编码的缺点：①计算的机率必须附加在编码后的文字上，这使得整个结果加长；②计算的机率是整个文字的机率，因此无法对部分地区的有序数列进行优化。

常见的熵编码有：香农算法[17,18](Shannon-Fano-Elias)、算术编码[19,20](Arithmetic coding)和哈夫曼[21](Huffman)编码。

Shannon-Fano-Elias 是一种基于一组符号集构建前缀码的技术。在理想意义上，其与 Huffman 编码一样，并未实现编码词长度的最低预期；然而，其确保了所有的编码词长度在一个理想的理论范围 $-\log_2 \frac{1}{P(x)}$ 之内。

若使用 k 进位的编码，Shannon 编码定理如下：

$$\frac{E}{\log_2 k} \leqslant m(L) \leqslant \frac{E}{\log_2 k} + 1 \tag{1}$$

但是 Shannon-Fano-Elias 并不总是产生最优的前缀码，所以这种方法不适合应用于抽油机示功图数据

的压缩。

算数编码和其他熵编码方法不同的地方在于，其他的熵编码方法通常是把输入的消息分区为符号，然后对每个符号进行编码，而算术编码是直接把整个输入的消息编码为一个数，一个满足($0 \leqslant n < 1.0$)的小数 n。对于抽油机示功图数据是 500 个重复度很小的实数，很显然不能使用算数编码这种方式来进行数据压缩。

经过大量样本数据分析，发现最适用于示功图数据的压缩方法为 Huffman 提出的编码方法即 Huffman 算法[22]。

哈夫曼树(Huffman Tree)[23]，又称最优树，是一种带权路径长度(WPL)最小的二叉树。其基本术语为：①结点之间的路径和路径长度，从树中一个结点到另一个结点之间的分支构成这两个结点之间的路径，路径上分支的数目称作这两个结点之间路径长度；②结点的权和带权路径长度，从根结点到结点的路径长度与结点的权的乘积；③树的带权路径长度，树中所有叶子结点的带权路径长度之和。

$$\mathrm{WPL} = \sum_{i=1}^{n} w_i L_i \tag{2}$$

Huffman 算法输入：

符号集合 $S = \{S_1, S_2, \cdots, S_n\}$，其 S 集合的大小为 n；权重集合 $W = \{W_1, W_2 \cdots, W_n\}$，其 W 集合不为负数，且 $W_i = \mathrm{weight}(S_i)$，$1 \leqslant i \leqslant n$。

Huffman 算法输出：

一组编码 $C(S,W) = \{c_1, c_2, \cdots, c_n\}$，其 C 集合是一组二进制编码且 c_n 为 S_n 相对应的编码，$1 \leqslant i \leqslant n$。

目标：

设 $L(C) = \sum_{i=1}^{n} w_i L(c_i)$ 为 C 的加权路径长，对所有编码 $T(S,W)$，则 $L(C) \leqslant L(T)$。

熵：

$$E = \sum_{j=1}^{J} P(S_j) \cdot \log_2 \frac{1}{P(S_j)} \tag{3}$$

Huffman 平均长度：

$$m(L) = \sum_{j=1}^{J} P(S_j) \cdot L(S_j) \tag{4}$$

根据 Shannon 编码定理，设 $b = \mathrm{mean}(L) \cdot N$，Huffman 平均编码长度为：

$$N \frac{E}{\log_2 k} \leqslant b \leqslant N \frac{E}{\log_2 k} + N \tag{5}$$

Huffman 编码是一种无前缀编码，且保证 WPL 最小。经过约 2×10^4 幅抽油机示功图的研究，Huffman 编码可以对示功图数据进行压缩，在样本数据重复性良好的情况下，有一定的效果，但是该方法需要存储数据字典，即需要编写一本解释各种代码意义的“词典”，即码簿，那么就可以根据码簿逐码依次进行译码。若每幅抽油机示功图存储一个数据字典，则需要大量的空间存储此字典就大大降低了空间压缩的可能性。若通过数据分析，得到统一的数据字典作为配置文件进行处理可以有效节约存储字典的空间；但是会出现数据字典通用性的问题，如果示功图数据的差异性较大，那么数据字典的长度就会非常长，使数据压缩效果受到影响。经过约 2×10^4 幅示功图的分析得出每个示功图数据大约需要 9 位。

综上，虽然 Huffman 编码的压缩算法可以使示功图的数据进行压缩，但是这种算法如果用于实际生产，其压缩效果不够理想，成本依然偏高。

2.2 Dynachart Compression

为了解决传统数据压缩算法应用于抽油机示功图数据压缩的不足，经过对大量示功图数据的研究，创造性提出了一种可以将抽油机示功图数据无损压缩的存储算法，即 Dynachart Compression (Dynamometer card Compression)，这种算法可以将示功图数据压缩存储到实时数据库中。

2.2.1 基本概念

解决抽油机示功图的数据压缩问题，首先需要解决示功图数据重复性低的问题。本文通过约 2×10^4 幅示功图数据的研究，对其规律进行了分析，使用了包括线性拟合、均值、差值等各种方法寻找规律，并经过对比分析，发现示功图相邻数据的差值的重复性最高。此方法的基本概念如下：

(1) 源点数据(SD，Source data)：第一个载荷或位移数据。

(2) 差值(DV，Different Value)：从左至右逐个计算相邻两个数值的差，且使用整数存储。

(3) 正常差值(NDV，Normal Different Value)：根据样本数据分析得到正常差值，即出现频度较高的差值数据。

(4) 特殊差值(SDV，Special Different Value)：DV 与 NDV 的差集合。

2.2.2 数据处理

Dynachart Compression 中的数据处理步骤如下：

(1) 去除错误的采集数据。根据样本数据分析，大约有 3.9/1 000 的数据是由于示功仪故障采集的错误数据，这些错误数据可以通过绘图分析得到，通常错误数据的特性是图形拟合后不呈现正常示功图，为提高数据压缩效率和保存精度，可以将这些错误数据予以清除。

(2) 确认数据精度。针对大量示功图数据分析，

确认可以接受的精度，按照此精度可以对示功图数据进行处理。一组示例示功图如下所示，图 5 是精度确认前的图形，图 6 是精度确认后的图形。

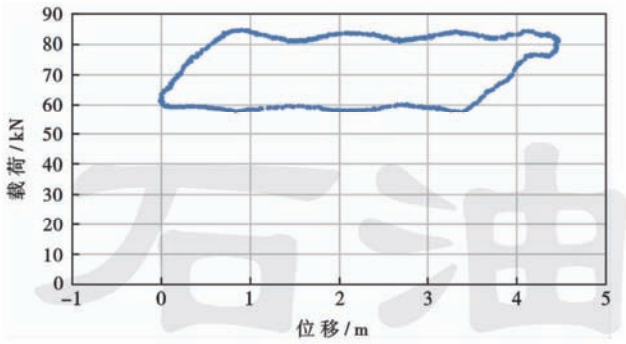

图 5 初始示功图

Fig. 5 Initial dynamometer

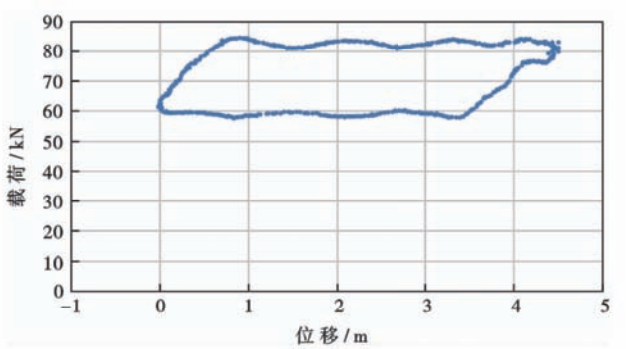

图 6 经过精度处理后的示功图

Fig. 6 Dynamometer after precision processing

(3) 获取示功图数据差值。除 SD 外，对 NDV、SDV 进行计算并记录。

(4) 按照 SD、NDV 和 SDV 的不同属性分别对这些数据进行存储。第一个字节先存储源点数据 SD，然后存储特殊差值 SDV 的个数，再依次存储每个特殊差值的位置和数据值，最后按位依次存储所述正常差值 NDV，其中每个正常差值的首位是符号位。所述的按位存储即将表示示功图正常差值数据的特定位数的二进制数据逐个比特进行存储，例如正常差值 13 可以用五位二进制 01101 表示，其中第一位“0”为符号位，表示该正常差值 13 为一个正值。此数据存储方式可以在保障数据信息无损和可还原性的基础上压缩示功图数据(图 7)。

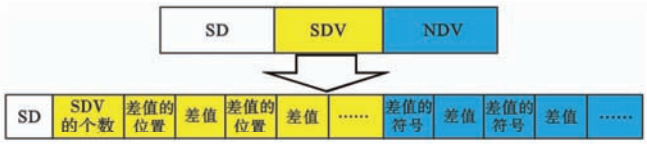

图 7 存储算法

Fig. 7 Storage algorithm

此算法需要的存储空间为：

$$M = 5 + \frac{\sum_{i=1}^{n}[P_{SDV(i)} + L_{SDV(i)}]}{8} + \frac{\sum_{i=1}^{n} L_{NDV(i)}}{8} \quad (6)$$

(5) 在实时数据库中，最大容量的存储方式是按照字符串进行存储，每个字符串可以存储 128 个字节。根据上述方法的描述，在样本中载荷数据最坏情况下需要 181.25 字节，位移数据需要 74.25 字节。因此 3 个实时库点的存储空间就能存下一幅示功图的样本数据。经过大量样本数据的分析，发现位移数据重复性较高，所以使用空间共享的原理，将数据进一步进行压缩，可将一幅抽油机示功图数据存入 2 个实时库点中(图 8)。

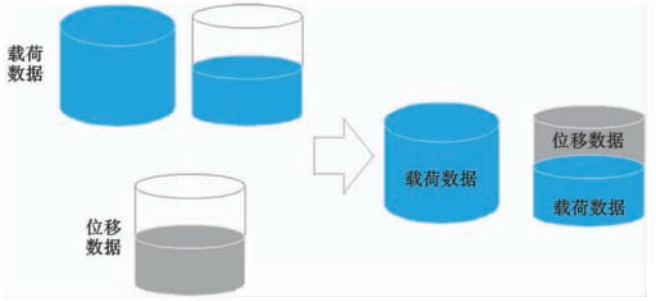

图 8 数据存储

Fig. 8 Data storage

2.2.3 实 例

以一组示功图数据来说明 Dynachart Compression 的存储过程。

载荷数据：76.46，76.24，75.77，75.03，73.89，72.51，70.94，69.42，68.1，67.22，66.72，66.66，66.82，67.18，67.83，68.72，69.84，71.12，72.4，73.45，74.14，74.46，74.42，74.08，…，75.96，76.33；共 250 个实数。

位移数据：0.26，0.29，0.33，0.36，0.39，0.43，0.46，0.5，0.54，0.58，0.62，0.67，0.71，0.75，0.81，0.85，0.9，0.94，0.99，1.04，1.1，1.15，1.2，1.25，…，0.21，0.23；共 250 个实数。

如果将其存储于实时数据库，每个实数需要 1 个数据点，那么共需要 500 个实时数据点。

使用 Dynachart Compression 方法，对载荷数据进行如下处理：

精度处理后：76.5，76.2，75.8，75，73.9，72.5，70.9，69.4，68.1，67.2，66.7，66.7，66.8，67.2，67.8，68.7，69.8，71.1，72.4，73.5，74.1，74.5，74.4，74.1，…，76，76.3。

取差值：0.3，0.4，0.8，1.1，1.4，1.6，1.5，1.3，0.9，0.5，0，−0.1，−0.4，−0.6，−0.9，−1.1，−1.3，

-1.3，-1.1，-0.6，-0.4，0.1，0.3，…，-1.7，-0.3。

扩大 10 倍以方便存储：3，4，8，11，14，16，15，13，9，5，0，-1，-4，-6，-9，-11，-13，-13，-11，-6，-4，1，3，…，-7，-3。

存储方法：前 4 个字节，76.5；第 5 个字节，1（SDV 的数量）；第 6 个字节，6（SDV 的位置）；第 7 个字节，16（SDV 的值）；从第 8 个字节开始按位存储 NDV，0001100100010000101101110011110110101001001010000010001101001011011001110111110111011011011010100000100011…1011110011。

根据式（6），需要 162 个字节。

位移数据进行精度处理后：0.3，0.3，0.3，0.4，0.4，0.4，0.5，0.5，0.5，0.6，0.6，0.7，0.7，0.8，0.8，0.9，0.9，0.9，1，1，1.1，1.2，1.2，1.3，…，0.2，0.2。

取差值：0，0，-0.1，0，0，-0.1，0，0，-0.1，0，-0.1，0，-0.1，0，-0.1，0，0，-0.1，0，-0.1，-0.1，0，-0.1，…，0，0。

扩大 10 倍以方便存储：0，0，-1，0，0，-1，0，0，-1，0，-1，0，-1，0，-1，0，0，-1，0，-1，-1，0，-1，…，0，0。

源点数据：0.3。

特殊差值：Null。

正常差值：0，0，-1，0，0，-1，0，0，-1，0，-1，0，-1，0，-1，0，0，-1，0，-1，-1，0，-1，…，0，0。

按照字符串类型存储此值，每个字符串可以最多存储 128 个字符：前 4 个字节，0.3；第 2 个字节，0（特殊差值的个数为 0）；从第 3 个字节开始按位存储 NDV，0000001010000001010000001010001010001010001010000001010001011010001 01…000000。

根据 Dynachart Compression 的计算公式，需要 67.25 个字节。

在这个实例中，共需要 162 + 67.25 = 229.25 个字节对数据进行存储。实时数据库每个数据点最多可以存储 128 个字节，可以将其存入 2 个实时数据库的数据点中。

3　应用效果

抽油机示功图数据无损压缩存储方法经过了数万幅示功图数据的实际验证，结合中国石油油气生产物联网系统项目的建设，已在西部地区 3 个油田、6 个作业区共计 891 口抽油机井的数据采集中投入使用。经过实际应用，该方法大大节省了存储空间和传输带宽，达到了降低建设成本、提高存储效率的目的。

此算法与 Huffman 算法效果的对比如下：

根据 Dynachart Compression 压缩公式，在最好情况下载荷数据需要 160.625 字节，最坏情况下需要 181.25 字节；位移数据在最好情况下需要 67.25 字节，最坏情况下需要 74.25 字节。Haffman 算法压缩的载荷数据在最好情况下需要 33.125 字节，最坏情况下需要 282.125 字节；位移数据在最好情况下需要 33.125 字节，最坏情况下需要 282.125 字节。Haffman 的最好情况是要求所有示功图数据都是相同数据的情况，这种情况是不可能存在的，而最坏情况即所有抽油机示功图数据都不相同的情况却经常出现。所以 Dynachart Compression 的平均情况要远好于 Haffman 的结果。

油田应用前、后效果对比如下：

（1）使用此算法前需要配置组态软件服务器、关系数据库服务器、实时数据库服务器、关系数据库软件、实时数据库软件、组态软件；使用本算法后无需配置关系数据库服务器和关系数据库软件。

（2）示功图使用此算法前一个作业区需要安装一台服务器，如果中国石油全部实施共需购置服务器及数据库软件 400 多台套，使用此算法后无需购置。

（3）使用此算法前，每个作业区需要 3 台服务器的电能、人力资源和维护成本；使用此算法后仅需要 2 台服务器的相关成本。

（4）存储空间方面（按 200 口井存储 3 年进行计算），使用此算法前需要存储空间 31.54G，使用后仅需要 2.69G。

（5）数据传输对网络资源的占用方面，使用此方法前每幅示功图需要传输 500 个数据点，使用此方法后仅需要 2 个数据点。

实践证明该方法可以实现抽油机示功图数据无损压缩至实时数据库，且使用的实时数据库点数是可以接受的。

4　结　论

（1）针对抽油机示功图数据压缩至实时数据库数据量过大、费用过高的问题，通过对中国石油各油气田几万幅示功图进行的数据分析提出了一种压缩算法 DynaChart Compression。此方法可以将每组示功图数据从常规的 500 个数据点压缩至 2 个实时数据库点中，压缩后的数据解压后可以满足工况诊断、产量计量等需求，比传统的关系数据库存储法节约空间、部署费用和维护成本。

（2）为了保证本方法的通用性，将会在进一步的工作中对更多井的示功图数据进行分析，完善样本资料，精确配置数据。该方法具有可扩展性，若发现更大的正常差值范围，可以使用扩展正常存储位升级示功

图数据实时存储压缩算法，每扩展一位正常差值的存储位数可以将现在可存储的数据范围加倍。

符号注释： S_{pe}—柱塞有效冲程，m；S_p—冲程，m；$P(x)$—变量 x 出现的概率；k—进制位数；E—熵；$m(L)$—编码长度；WPL—带权路径长度；n—叶子总数；w_i—权重；L_i—路径长度；$P(S_j)$—S_j 在 S 中出现的概率；$L(S_j)$—S_j 的编码长度；N—数据长度；L—相应差值分配的位数，bit；$L_{SDV(i)}$—存储第 i 个 SDV 所占用的空间，bit；$L_{NDV(i)}$—存储第 i 个 NDV 所占用的空间，bit；n—特殊差值的个数；P—存储 SDV 的位置所占用的空间，bit；$P_{SDV(i)}$—存储第 i 个 SDV 所在位置所占用的空间，bit；M—总存储空间，B。

参 考 文 献

[1] 陈元千，郝明强. HCZ 模型在多峰预测中的应用[J]. 石油学报，2013，34(4)：747-752.
Chen Yuanqian, Hao Mingqiang. Application of HCZ model to predicting multiple production peaks[J]. Acta Petrolei Sinica, 2013, 34(4): 747-752.

[2] 陈元千，邹存友. 预测油田产量和可采储量模型的典型曲线及其应用[J]. 石油学报，2014，35(4)：749-753.
Chen Yuanqian, Zou Cunyou. Model's typical curve and its application for forecasting production and recoverable reserves of oilfields[J]. Acta Petrolei Sinica, 2014, 35(4): 749-753.

[3] 朱荣杰，张国庆. 抽油机井故障诊断及处理方法[M]. 北京：石油工业出版社，2011：22-69.
Zhu Rongjie, Zhang Guoqing. Fault diagnosis and processing method for well pumping[M]. Beijing: Petroleum Industry Press, 2011: 22-69.

[4] Kang K D, Son S H, Stankovic J A. Specifying and managing quality of real-time data services[R]. Charlottesville, VA: University of Virginia, 2002.

[5] 何岩峰，吴晓东，韩国庆，等. 示功图频谱分析新方法[J]. 石油学报，2008，29(4)：619-624.
He Yanfeng, Wu Xiaodong, Han Guoqing, et al. Frequency spectrum analysis method for recognition of dynamometer card[J]. Acta Petrolei Sinica, 2008, 29(4): 619-624.

[6] Sayood K. 数据压缩导论[M]. 第 4 版. 贾洪峰，译. 北京：人民邮电出版社，2014：1-138.
Sayood K. Introduction to data compression[M]. 4th ed. Jia Hongfeng, trans. Beijing: Posts & Telecom Press, 2014: 1-138.

[7] 吴乐南. 数据压缩[M]. 第 3 版. 北京：电子工业出版社，2012：1-7，27-44.
Wu Lenan. Data compression[M]. 3rd ed. Beijing: Publishing House of Electronics Industry, 2012: 1-7, 27-44.

[8] Hilbert M, López P. The world's technological capacity to store, communicate, and compute information[J]. Science, 2011, 332(6025): 60-65.

[9] Witten I H, Bell T C, Moffat A, et al. Semantic and generative models for lossy text compression[J]. The Computer Journal, 1994, 37(2): 83-87.

[10] Mahmud S. An improved data compression method for general data[J]. International Journal of Scientific & Engineering Research, 3(3): 1-4.

[11] Chanda P, Elhaik E, Bader J S. HapZipper: sharing HapMap populations just got easier[J]. Nucleic Acids Research, 2012, 40(20): e159.

[12] Salomon D, Motta G. Handbook of data compression[M]. London: Springer, 2010.

[13] Gagie T, Gawrychowski P, Puglisi S J. Approximate pattern matching in LZ77-compressed texts[J]. Journal of Discrete Algorithms, 2015, 32: 64-68.

[14] Ziv J, Lempel A. Compression of individual sequences via variable-rate coding[J]. IEEE Transactions on Information Theory, 1978, 24(5): 530-536.

[15] Welch T A. A technique for high-performance data compression[J]. Computer, 1984, 17(6): 8-19.

[16] Nelson M R. LZW data compression[J]. Dr. Dobb's Journal, 1989, 14(10): 29-36.

[17] Adjeroh D, Nan Fei. Suffix-sorting via shannon-fano-elias codes[J]. Algorithms, 2010, 3(2): 145-167.

[18] Shannon C E. A mathematical theory of communication[J]. ACM Sigmobile Mobile Computing and Communications Review, 2001, 5(1): 3-55.

[19] MacKay D J C. Information theory, inference and learning algorithms[M]. Cambridge: Cambridge University Press, 2003.

[20] Witten I H, Neal R M, Cleary J G. Arithmetic coding for data compression[J]. Communications of the ACM, 1987, 30(6): 520-540.

[21] Wan S H, Huffman D H, Azarnoff D L, et al. Effect of route of administration and effusions on methotrexate pharmacokinetics[J]. Cancer Research, 1974, 34(12): 3487-3491.

[22] Huffman D A. A method for the construction of minimum-redundancy codes[J]. Proceedings of the IRE, 1952, 40(9): 1098-1101.

[23] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms[M]. 3rd ed. Cambridge: The MIT Press, 2009.

（收稿日期 2015-07-30 改回日期 2016-01-13 编辑 宋 宁）