实时数据库有损压缩算法的研究

杨永军[1],徐 江[1],许 帅[2],舒 逸[2]
(1. 上海麦杰科技股份有限公司,上海 200233;
2. 上海交通大学,上海 200240)

摘 要:在实时数据库中,测点数量多,数据量庞大,数据变化慢,数据冗余多,且实时数据库对实时性的要求很高,因此需要高效的压缩算法对实时数据进行压缩。实时数据库中的数据压缩算法分为有损和无损两类,文中就数据有损压缩进行了研究。通过对现有的有损压缩算法进行分析和比较,总结并提出了一个新的算法。该算法基于预测和动态修正,对实时数据进行快速高效的有损压缩。通过测试和比较,该算法在提高压缩比的同时能满足系统对还原精度的要求。

关键词:实时数据库; 数据压缩; 有损压缩
中图分类号:TP301.6 **文献标识码**: A **文章编号**:1673-629X(2012)09-0005-04

Research on Lossy Compression Algorithm in Real-time Database

YANG Yong-jun[1], XU Jiang[1], XU Shuai[2], SHU Yi[2]
(1. Shanghai Magus Technology Co., Ltd., Shanghai 200233, China;
2. Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: In real-time database, amount of measuring points and size of data are very huge. The data varies slowly with lots of redundancy, and the system is of high demand for efficiency. So that efficient algorithm of compression is one of the most important aspects of real-time database system. There're two types of compression algorithm, lossy and lossless ones. It studied data lossy compression and proposed a new algorithm by analyzing and comparing existing lossy compression algorithms. It's based on prediction and dynamic correction to be lossy compression quickly and efficiently for real-time data. Through testing, the algorithm improves the compression rate and decompression accuracy at the same time.

Key words: real-time database; data compression; lossy compression

0 引 言

实时数据库主要被用于工业监控领域,如火电厂厂级监控系统(SIS)。在工业监控领域,数据库应用有以下特点[1]:

(1)测点数量多。一个新建 300WM 的火电厂的厂级监控系统测点数量超过 10000 个。且测点数量多意味着系统并发任务多,因此对系统运行效率要求高;

(2)数据量庞大。工业监控系统的数据采样频率多为秒级,即每秒每个测点获得一个采样数据,如果按上述 300WM 的火电厂的 10000 个测点计算,系统每秒需要处理和保存 10000 个数据。因此数据处理算法必须在有限的时间内完成数据处理才不会造成瓶颈。另一方面,庞大的数据量需要大量硬盘空间用于存储。

观察工业监控的实测数据,可见其数据有以下特点:

(1)数据变化慢,冗余多。因为采样频率高,所以工业监控系统的实时数据多有变化慢的特点,特别对于整型与开关型变量,有大量冗余存在;

(2)与时间相关性强。工业监控系统的数据随时间推移而变化,数据值与时间有很强的相关性;

(3)存在不可避免的噪声。工业监控系统的数据在测量与传输过程中都有可能引入噪声。

因此数据压缩算法是实时数据库中的关键技术,数据压缩的主要目的有两方面[2]:

一是减小存储体积,减少存储成本;

二是减少网络传输过程中占用的带宽,提高网络带宽利用率,减少网络传输时间。

数据压缩可分为有损压缩和无损压缩两种[3]。而实时数据库中多采用先有损压缩,后无损压缩的二次压缩以达到更高的压缩比[4]。有损压缩的原理是,通过一定的数据筛选规则,选择性地保留部分数据,抛弃部分数据,并保证解压算法的还原精度在系统精度误差允许范围内[5]。因此有损压缩算法有较高的压缩

收稿日期:2012-01-25;**修回日期**:2012-04-29
基金项目:工信部电子信息产业发展基金(工信部财[2009]453 号)
作者简介:杨永军(1972-),男,湖北潜江人,硕士,工程师,研究方向为实时数据库系统和生产信息化系统。

比,但同时会导致系统精度下降。

文中就实时数据库中数据的有损压缩算法进行了深入的研究,并提出了一个新的算法,可以在提高压缩比的同时满足还原精度的需求。

1　有损压缩算法

现有的有损压缩方法可以分为两大类:死区压缩和趋势压缩[6]。

在60到70年代,因为空间遥测技术的需要,研究者们开始研究数据压缩算法[7]。80年代早期,Hale和Sellars提出了死区(boxcar)和反向斜率(backslope)压缩算法[8],通过预测的方法对数据进行压缩,并被用于历史数据监控领域。

趋势压缩则以旋转门[9]算法为代表,旋转门算法大大提高了实时数据库的压缩率,被广泛应用在实时数据库领域[10,11]。但因为旋转门算法并没有对趋势的变化情况加以描述、记录和预测,因此压缩率和精确度都仍有提升的空间。

国内对于有损压缩算法的研究主要以图像和视频的有损压缩算法为主,如矢量量化编码算法[12],基于小波变换的一系列压缩算法[13,14],且主要被应用于医学领域[15]。

Kortman提出了扇域内插算法(Fan Interpolators)[16],该方法被Peter A. James简称为SLIM(Straight-Line-Interpolative-Methods,直线内插方法)算法,后者于1995年改进SLIM算法并提出了SLIM2和SLIM3算法[17],该算法的压缩率和精确度较旋转门算法都有一定的优势,但是并未在实时数据库系统中被引入使用,且该算法仍有改进的空间。

2　基于预测和动态修正的有损压缩算法

2.1　关于有损压缩算法和拟合函数

通过研究和观察,发现不同的有损压缩算法可以根据其解压算法对应的拟合函数的阶数进行分类。

死区压缩算法,其对应的拟合函数为简单的:

$$y' = y$$

对于时间t,该函数为零次函数,差值小于误差容忍范围的值都被忽略,解压时按前次记录点的值计算,且解压算法与时间无关。

反向斜率压缩算法、旋转门算法及SLIM算法,解压算法均对应拟合函数:

$$y' = y_2 + k(t' - t_2)$$

$$k = (y_2 - y_1)/(t_2 - t_1)$$

对于时间t,该函数为一次函数,因此这三种算法的拟合度都比死区压缩算法高,从而得到了更高的压缩比和更高的还原精度。旋转门算法和SLIM算法通过在检验时将值的比较换成斜率的比较,在保证效率的同时提高了压缩比。而SLIM2算法通过对保存值的修改,对值进行了简单的趋势预测,但是这牺牲了还原精度。

如果用二次线性拟合函数作为解压算法,则会导致性能的大幅降低。因此,文中提出了一种折衷的方案,通过预测和动态修正,使得压缩效果接近于二次线性拟合而压缩效率没有明显的下降。

2.2　压缩算法

文中提出的压缩算法的数据筛选过程基于SLIM的扇形插值法,在此基础上,通过预测和动态修正,大幅提高了压缩比,也提高了还原精度。

下面是文中提出的压缩算法的过程,假设误差容忍范围为 $\pm T$。

2.2.1　初始化

系统初始化过程包括记录最初两个数据 $p_1(t_1, v_1)$,$p_2(t_2, v_2)$,计算初始时的扇形区域斜率上限:

$$r_{up} = (v_2 + T - v_1)/(t_2 - t_1)$$

斜率下限:

$$r_{down} = (v_2 - T - v_1)/(t_2 - t_1)$$

斜率中限:

$$r = (v_2 - v_1)/(t_2 - t_1)$$

初始化斜率修正值d_r为0,将p_1作为"最后写入数据"记录为$p_{last}(t_{last}, v_{last})$,$p_2$作为"最后读入数据"记录为$p_{read}(t_{read}, v_{read})$。

2.2.2　实时读取和计算数据

在系统运行过程中,不断读取新的实时数据,对于读入的新数据$p(t,v)$,计算其对应的斜率上限:

$$r'_{up} = (v + T - v_{last})/(t - t_{last}) + d_r * (t - t_{last})$$

斜率下限:

$$r'_{down} = (v - T - v_{last})/(t - t_{last}) + d_r * (t - t_{last})$$

2.2.3　比较数据确定是否保存

如果$r'_{up} > r_{up}$且$r'_{down} > r_{up}$,那么需要保存新数据;如果$r'_{down} < r_{down}$且$r'_{up} < r_{down}$,也需要保存新数据,否则,不保存新数据,只重新调整参数。

2.2.4　保存数据并调整参数

根据2.2.3的结果,如果需要保存新数据,则计算斜率修正值

$$d_r = ((v - v_{read})/(t - t_{read}) - r)/(t_{read} - t_{last})$$

做一通过"最后读入数据"的纵垂线,如果$r'_{up} > r_{up}$,把该纵垂线与r_{up}的交点写入数据库并记录为"最后写入数据",如果$r'_{down} < r_{down}$,把该纵垂线与r_{down}的交点写入数据库并记录为"最后写入数据",重新计算:

$$r_{up} = (v + T - v_{last})/(t - t_{last}) + d_r * (t - t_{last})$$

$$r_{down} = (v - T - v_{last})/(t - t_{last}) + d_r * (t - t_{last})$$

若无需保存新数据,如果 $r'_{up} < r_{up}$,调整 r_{up} 为 r'_{up},如果 $r'_{down} > r_{down}$,调整 r_{down} 为 r'_{down}。无论是否保存新数据,记录当前数据为"最后读入数据"。

下面是压缩算法核心部分的伪代码:

```
while ture do
读取新的数据对(t, v)
  if 未初始化 then
    t1←t, v1←v, tlast←t, vlast←v
读取新的数据对(t, v)
    t2←t, v2←v, treat←t, vread←v
rup←(v2+T-v1)/(t2-t1)
rdown←(v2-T-v1)/(t2-t1)
r←(v2-v1)/(t2-t1)
dr←0
  else
    rup2←(v+T-vlast)/(t-tlast)+dr*(t-tlast)
    rdown2←(v-T-vlast)/(t-tlast)+dr*(t-tlast)
if (rup2>rup and rdown2>rup) or (rup2<rdown and rdown2<
rdown) then
dr←((v-vread)/(t-tread)-r)/(tread-tlast)
      if rup2>rup then
        vnew←vlast+rup(t-tlast)
      else
        vnew←vlast+rdown(t-tlast)
      endif
      vlast←vnew, tlast←tread
      保存(vlast, tlast)至数据库
      rup←(v+T-vlast)/(t-tlast)+dr*(t-tlast)
      rdown←(v-T-vlast)/(t-tlast)+dr*(t-tlast)
else
    if rup2<rup then rup←rup2
    if rdown2>rdown then rdown←rdown2
endif
vread←v, tread←t
endif
repeat
```

2.3　解压算法

文中提出的压缩算法对应的解压算法如下。对于输入的查询时间段,从数据库读取对应的数据集,为了减小网络传输数据量,数据拟合工作则在客户端完成。在客户端进行拟合时,对于时间 t,读取 t 之前最近的 3 个数据对 $p_1(t_1, v_1)$,$p_2(t_2, v_2)$,$p_3(t_3, v_3)$,计算 $p_t(t, v_t)$ 的方法如下:

$$k_1 = (v_2 - v_1)/(t_2 - t_1)$$

$$k_2 = (v_3 - v_2)/(t_3 - t_2)$$

$$v_t = v_3 + (t_3 - t_2)(k_2 + (k_2 - k_1)(t_3 - t_2)/(t_2 - t_1))$$

由于输出结果的拟合在客户端执行,因此,该过程不会影响服务器端的工作效率。

3　算法效率

文中对旋转门、SLIM、SLIM2 和新的算法从理论效率和实际效率两方面进行了测试和比较。

3.1　理论效率

可以根据各个算法的数学模型对理论效率进行测试和评估。选取正弦函数 $y[i] = 100\sin(t[i])$ 作为测试用函数,理论效率的估算值如表 1。

从 $t = 0$ 开始,根据各有损压缩算法的规则,按每秒 1 次的采样频率进行采样,误差容忍范围为 ±1.5。

表 1　有损压缩的理论效率测试

算法	压缩比	还原误差
未压缩	1	0
旋转门	17.8	1.48
SLIM	35	1.5
SLIM2	22.1	1.17
文中提出的算法	28.5	1.15

从表中可以看到,文中提出的算法从理论上来说比旋转门和 SLIM2 要好。

3.2　实际效率

实测数据为带噪声的正弦函数 $y = 100\sin(t) + G$,其中 G 为随机噪声,对于该函数,实际测试的平均结果如表 2 所示:

表 2　有损压缩算法的实际效率测试

算法	压缩比	还原误差
未压缩	1	0
旋转门	10.1	1.5
SLIM	16.7	1.5
SLIM2	5.3	1.32
文中提出的算法	7.8	1.28

从上表可以看出,对于带有噪声的实际数据,文中提出的算法在保证压缩比的前提下提高了还原精度。但是由于噪声的加入,预测值与实际值的偏差增大,因此压缩比与理论值相比降低了很多。

4　结束语

数据压缩算法是实时数据库中最重要的系统参数之一,区别于传统数据库的数据压缩,在用于工业监控领域的实时数据库中,有损压缩是区别于无损压缩的一个重要环节。文中就实时数据库中的有损数据压缩,提出了一个新的基于动态的预测和修正的有损压缩算法,在提高压缩比的同时,对还原精度的影响也在误差容忍范围以内,为该领域的研究做出了贡献。今后工作将专注于压缩算法的并行化[18]和分布式存储[19],以进一步提高系统的性能。

参考文献:

[1] Singhal M. Issues and approaches to design of real-time database systems[J]. ACM SIGMOD Rec. ,1988,17(1):19-33.

[2] Cappellini V. A study on data-compression techniques[C]// European Space Research Organisation. [s. l.]:[s. n.], 1974.

[3] Nelson M, Gailly Jean-Loup. The data compression book [M]. 2nd ed. New York:MIS Press,1995.

[4] Iyer B R, Wilhite D. Data compression support in databases [C]//Proceedings of the 20th International Conference on Very Large Data Bases. [s. l.]:[s. n.],1994:695-704.

[5] Salomon D. Data compression: the complete reference[M]. New York:Springer-Verlag New York, Inc. ,1997.

[6] Roth M A, van Horn S J. Database compression[J]. ACM SIGMOD Rec. ,1993,22(3):31-39.

[7] Ehrman L. Analysis of some redundancy removal bandwidth compression techniques[J]. Proceedings of the IEEE,1967, 55(3):278-287.

[8] Hale J C, Sellars H L. Historical data recording for process computers[J]. CEP,1981,77(11):38-43.

[9] Bristol E H. Swinging door trending: adaptive trend recording [C]//ISA National Conf. Proc. . [s. l.]:[s. n.],1990:749-754.

[10] Kennedy J P. Data treatment and applications future of the desktop[C]//Proceeding of Foundations of Computer-aided Process Operations. [s. l.]:[s. n.],1993:21-43.

[11] Mah R S H. Process trending with piecewise linear smoothing [J]. Computers & Chemical Engineering,1995,19(2):129-137.

[12] 陆哲明. 矢量量化编码算法及应用研究[D]. 哈尔滨:哈尔滨工业大学,2001.

[13] 万 刚,朱长青. 多进制小波及其在DEM数据有损压缩中的应用[J]. 测绘学报,1999,28(1):36-40.

[14] 田宝凤,徐抒岩,孙荣春,等. 一种适合星上应用的遥感图像有损压缩算法[J]. 光学精密工程,2006,14(4):725-730.

[15] 肖振国,田 心. 医学图像无损压缩与有损压缩技术的进展[J]. 国外医学(生物医学工程分册),2002,25(2):55-58.

[16] Kortman C M. Redundancy reduction-a practical method of data compression[J]. Proceedings of the IEEE,1967,55(3): 253-263.

[17] James P A. Data compression for process historians[C]// CAST Communications. [s. l.]:[s. n.],1996.

[18] 祝 君,林庆农,徐造林,等. 实时历史数据库中压缩技术的并行化研究[J]. 计算机技术与发展,2010,20(7):36-39.

[19] 陈建英,刘心松. 基于大规模分布式副本定位的分级索引压缩机制[J]. 电子科技大学学报,2011,40(4):554-558.

(上接第4页)

4 结束语

文中提出了一种基于CUDA的三维层位面自动追踪算法,跟原有的CPU版本的层位面自动追踪算法相比,计算速度有了明显的提升。但是由于文中采用的层位面自动追踪算法是一个不断迭代的过程,每一次迭代都需要进行显存和内存之间的数据拷贝。同时由于再执行调整操作时,不同CUDA线程的计算量并不完全相同,这就导致了thread divergence[12],这也限制了文中算法的速度。如何解决这两个问题是下一步工作的重点。

参考文献:

[1] 陆基孟. 地震勘探原理(上册)[M]. 东营:石油大学出版社,1993.

[2] Faraklioti M, Petrou M. Horizon picking in 3D seismic data volumes[J]. Machine Vision and Applications,2004,15(4): 216-219.

[3] 李雪峰,阎建国,赵 州,等. 利用相干属性剖面特征进行层位解释[J]. 物探化计算技术,2011,33(2):134-139.

[4] Bahorich M, Farmer S. The coherence cube[J]. The Leading Edge,1995,14(10):1053-1058.

[5] NVIDIA Corporation. NVIDIA CUDA C Programming Guide, Version 4.0[EB/OL]. 2011. http://developer. nvidia. com/nvidia-gpu-computing-documentation.

[6] Sanders J, Kandrot E. CUDA by Example: An Introduction to General-purpose GPU Programming[M]. 北京:清华大学出版社,2010.

[7] Deschizeaux B, Blanc Jean-Yves. Imaging Earth's Subsurface Using CUDA[C]//GPU Gems 3. [s. l.]: Addison Wesley Professional,2007.

[8] Keskes N, Zaccagnino P, Rether D, et al. Automatic extraction of 3-D seismic horizons[C]//Annual Meeting Expanded Abstracts. [s. l.]:SEG,1983:557-559.

[9] dGB Earth Sciences. OpendTect User Documentation Version 4.0[EB/OL]. 2009. http://opendtect. org/rel/doc/User/base/.

[10] Harris M. Parallel Prefix Sum with CUDA[C]//GPU Gems 3. [s. l.]:Addison Wesley Professional,2007.

[11] Harish P, Narayanan P J. Accelerating large graph algorithms on the GPU using CUDA[C]//High performance computing-HIPC 2007, lecture notes in computer science. [s. l.]:[s. n.],2007:197-208.

[12] Kirk D B, Hwu W W. Programming Massively Parallel Processors[M]. [s. l.]:Morgan Kaufmann, 2010:96-103.

实时数据库有损压缩算法的研究

作者： 杨永军， 徐江， 许帅， 舒逸
作者单位： 杨永军,徐江(上海麦杰科技股份有限公司,上海 200233)， 许帅,舒逸(上海交通大学,上海 200240)
刊名： 计算机技术与发展
英文刊名： Computer Technology and Development
年，卷(期)： 2012(9)