第33卷 第6期
2010年6月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 33 No. 6
June 2010

科学计算浮点数据的高性能无损压缩

何克晶
(华南理工大学计算机科学与工程学院 广州 510641)

摘 要 科学计算在科学界及工业界发挥着越来越重要的作用，所随之产生的科学数据也越来越多. 因二级存储(如硬盘)的读写速度通常较慢，庞大的数据量除了占据存储空间之外，还影响着系统性能. 文中通过系统研究浮点数据的特性，建立预测精度和压缩比之间的关系的理论体系. 通过利用科学数据之间的相关性，采用多种预测器以及高效熵编码方法，提出一种科学数据高性能无损压缩方法. 该方法既不需要使用者有关于原始数据的先验知识，也不需要使用者自行设计预测器. 通过与其他压缩方法进行比较，结果表明文中方法的压缩比远高于其他方法，并同时具有恒定的海量压缩吞吐量. 该方法已被应用于大规模颗粒动力学仿真系统的数据压缩.

关键词 科学计算；浮点数据；无损压缩；高性能
中图法分类号 TP391 **DOI号**：10.3724/SP.J.1016.2010.00966

High Performance Lossless Compression of Scientific Floating Data

HE Ke-Jing
(*School of Computer Science and Engineering, South China University of Technology, Guangzhou* 510641)

Abstract Scientific computing is playing more and more important role in the scientific research and in the industry, and enormous valuable scientific data are produced. Since the read/write speed of secondary storages (e.g. hard disks) is usually slow, besides occupying storage space, mass data also affect system performance. This paper studies the properties of floating-point data thoroughly, and establishes the theoretical relationship between the prediction precision and the compression ratio. By exploiting the interdependency between scientific data and making use of several predictor and high performance entropy encoding method, this paper proposes a high performance lossless compression method for scientific data. The method doesn't require users having priori knowledge about the data, nor is custom-designed predictor required. Comparing results with other methods, show the method has much higher compression ratio and constant high throughput. The method has been used for the compression of scientific data in large scale discrete element simulation of particle dynamics.

Keywords scientific computing; floating data; lossless compression; high performance

1 引 言

随着科学计算技术在科学界及工程界的广泛应用，所随之产生的科学数据也越来越多. 主要的科学数据类型包括输入数据、模型数据、检查点(checkpoint)数据和仿真结果. 记录模拟过程的仿真结果通常占据着最大的存储空间. 同时，因为大规模科学计算应用的执行时间通常较长且问题域较大，为了确保模拟过程的顺利执行，所需要存储到硬盘的检

收稿日期：2009-04-01；最终修改稿收到日期：2010-05-13. 本课题得到国家自然科学基金(10805019)、广东省自然科学基金(8451064101000083)和华南理工大学中央高校基本科研业务费专项基金(2009ZM0040)资助. **何克晶**，男，1983年生，博士，副教授，主要研究方向为科学计算与模拟、并行计算与分布式系统. E-mail：kejinghe@ieee.org.

查点数据量也非常大.因硬盘 I/O 操作通常较慢，庞大的数据量除了占据存储空间之外，还影响着系统性能.

为了减少科学数据所需的存储空间大小，许多研究者都借助压缩工具来对数据进行压缩.目前最广为采用的有两种方式：第 1 种方式在数据产生之后，再借助第三方的压缩工具进行压缩.第 2 种方式中，压缩算法作为模拟软件的数据过滤器(filter)，数据通过过滤器处理，直接产生压缩后的数据.不管哪种方式，借助的都是通用压缩算法.

科学数据与通常数据有着本质的不同.在模拟系统中，科学数据多是以高精度浮点数的形式存在，所产生的检查点数据和结果数据也多是一维或者多维的浮点数组.通用压缩算法对于文本的压缩效果很好，但无法充分利用科学数据本身以及浮点数本身的特性，从而导致最终的时间性能和空间性能较差.

本文提出一种科学数据高性能无损压缩方法，它通过利用科学数据之间的相关性以及浮点数本身的特性来进行预测编码，从而进行快速有效的科学数据压缩.该方法既不需要使用者有关于原始数据的先验知识，也不需要使用者自行设计预测器.该方法既可作为单独科学数据压缩工具的内核使用，也可作为模拟软件的高性能数据过滤器.通过与常用压缩工具进行比较，结果表明该方法在时间性能和空间性能上均优于通用算法.该方法已被应用于大规模离散元科学仿真系统的数据压缩.

本文第 2 节介绍相关研究；第 3 节给出该科学数据高性能无损压缩方法的详细介绍，包括系统框架和该方法的关键技术：预测器、熵编码等；第 4 节对该方法进行评估；第 5 节总结全文.

2　相关工作

目前对通用科学数据的压缩较多采用的还是通用压缩方法和工具，如 Gzip、Bzip2 等.有部分研究对科学模型以及网格(Mesh)数据的压缩进行了探索.其中最有代表性的为文献[1-2].这些研究往往通过对三维网格数据进行分析处理，从而对相应的几何数据和连通性数据进行去冗余压缩.Isenburg 等[3]针对网格数据多为浮点数的特点，设计了针对浮点数存储格式的网格数据压缩方法.随后，Lindstrom 等[4]通过使用 Lorenzo 预测器[5]，继续将该方法拓展到具有空间冗余性的浮点数据压缩.在本文的横向比较中，该方法简称 PLMI.该方法只适用于具有空间冗余性的空间数据，如二维图像、三维速度场数据等.并且使用时必须显式指定每个数据维的大小.

Engelson 等[6]使用外延预测器来压缩存储模拟数据.该方法对平滑变化的数据效果较好，对通用的科学数据压缩效果较差.Ratanaworabhan 等[7]根据前若干个数据，使用差分预测器进行浮点数的预测压缩，能达到 1.6 左右的几何平均压缩比.Burtscher 等[8-9]设计了高通量的浮点数据压缩方法，能达到 1.9 左右的几何平均压缩比.在本文的横向比较中，该方法简称 FPC.

相关研究均没有研究预测精度和压缩比之间的关系.本文通过系统分析，建立预测精度和压缩比之间的解析关系.并采用多种预测器以及高效熵编码方法，得到远高于相关研究的压缩比，并同时具有恒定的海量压缩吞吐量.

3　科学数据的高性能无损压缩方法

3.1　IEEE 浮点数结构

现今计算机系统中普遍采用的都是 1985 年的 IEEE 二进制浮点数算术标准(IEEE 754)[10].在该标准中，定义了被广泛使用的 32 位二进制浮点数(单精度)和 64 位二进制浮点数(双精度)的存储格式.在 2008 年新的 IEEE 浮点数算术标准(IEEE 754-2008)[11]中，新增定义了 128 位二进制浮点数、64 位十进制浮点数以及 128 位十进制浮点数，5 种基本浮点数格式中的 3 种如图 1 所示.其中两种 128 位的格式因篇幅关系并未列出.

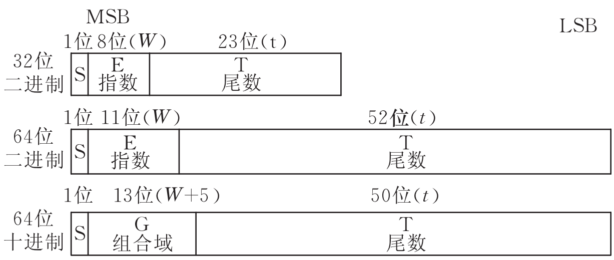

图 1　IEEE 二进制浮点数格式

对于二进制浮点数而言，数值由下式表示：

$$(-1)^S 2^{E-2^{W-1}+1}(1+T/2^t) \tag{1}$$

十进制浮点数的格式类似，只是在组合域(combination field)中，除了用 W 位来编码指数，其余的 5 位共 32 种组合中的 30 种组合用来编码归一化后的十进制数的整数部分(0～9 共 10 种取值)与 3 种指数值.剩余的 2 种组合分别表示无穷大与

NaN. t 位的尾数部分采用密集十进制(Densely Packed Decimal,DPD)编码 $3t/10$ 位十进制数的小数部分.

因 32 位和 64 位二进制浮点数最为常用,并且 IEEE 754-2008 标准同时定义了若干种交换格式用于在不同的浮点数实现中进行数据交换,所以本文以 32 位和 64 位二进制浮点数为例,将它们统称为浮点数.但本文的方法同样适用于其它的浮点数格式,甚至非标准 IEEE 格式.需要注意的是,IEEE 标准中并没有定义浮点数在机器中的字节存储顺序.有的采用小端法(little endian)进行存储;有的则采用大端法(big endian)进行存储.字节存储顺序并不影响压缩算法,但为了行文方便,本文均以小端法为例.

3.2 科学数据压缩原理

作为科学数据主体的浮点数据与文本数据等其他数据有诸多本质区别:

(1) 虽然浮点数本身是字节对齐的(byte-aligned),但各组成部分(如符号位、指数部分和小数部分)均不是字节对齐的.传统的基于字节的压缩方法不顾浮点格式的物理意义,而强行把数据分割成以字节为单位(图 2).因物理意义丢失,从而使得不管是字典法还是统计法,效果均不理想.

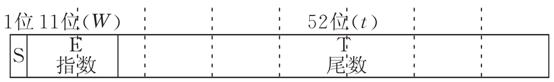

图 2 传统基于位的方法不能利用浮点数的格式特征

(2) 即使是以位为单位的预测方法,比如一些基于上下文的预测压缩方法[12-13],因无法考虑到浮点数各个位的意义并根据其重要性进行区别对待,从而导致需要处理的上下文过多,占用大量的内存空间和压缩时间,还不能得到较优的压缩比.

为了在快速压缩的同时得到高压缩比,压缩方法需要充分利用浮点数的特征.主要的特性包括:

(1) 指数部分的数值(E)可以基本代表该浮点数的范围.尾数部分对数值范围的影响度随着与 LSB 位距离的减小而指数降低.所以,某浮点数的大范围基本上用符号位 S、指数 E 及少数几个尾数位即能确定.也就意味着,可以用最靠近 MSB 的若干位,来近似整个浮点数.

定义 1. 区间.对浮点数 F,定义 $(-1)^S E_F - S$ 为该浮点数所位于的区间.对于两个二进制浮点数 F_1、F_2,如果符号位 $S_{F_1}=S_{F_2}$,并且指数位 $E_{F_1}=E_{F_2}$,则称 F_1 与 F_2 属于同一个区间.同时,把它们的尾数分别简记为 T_{F_1} 与 T_{F_2}.若 F_1、F_2 位于不同区间,则将 F_1 与 F_2 的区间之差定义为两者之间的区间距离.

定义 2. 前导零数.数值串 N 中第一个非零字符前面的零字符的个数,简记为 $LZC(N)$.

定理 1. 如果两个浮点数 F_1、F_2 位于同一个区间,记二进制异或运算为 $\oplus$,则平均的 $LZC(F_1\oplus F_2)$ 约等于 $W+2$.

证明. $\forall F_1$,因 F_2 与其位于同一区间,根据定义 1,可知它们的前 $W+1$ 位均相同.F_2 的尾数部分共 t 位,共有 2^t 种可能性.其中有 1 种可能性是 $LZC(T_{F_1}\oplus T_{F_1})=t(F_1=F_2$时$)$,$2^{t-x-1}$ 种可能性为 $LZC(T_{F_1}\oplus T_{F_1})=x(x=\{0,2,\cdots,t-1\})$.假设这 2^t 种可能性出现的概率一样,则平均而言:

$$\begin{aligned}\overline{LZC(T_{F_1}\oplus T_{F_2})}&=\frac{t\times 1+\sum\limits_{x=0}^{t-1}x\times 2^{t-x-1}}{2^t}\\&=\frac{t+\sum\limits_{x=1}^{t-1}\sum\limits_{i=x}^{t-1}2^{t-i-1}}{2^t}=\frac{t+\sum\limits_{x=1}^{t-1}(2^{t-x}-1)}{2^t}\\&=\frac{t+2^t-2-(t-1)}{2^t}=\frac{2^t-1}{2^t}\approx 1\end{aligned}\tag{2}$$

从而

$$\overline{LZC(F_1\oplus F_2)}=W+1+\overline{LZC(T_{F_1}\oplus T_{F_2})}\approx W+2\tag{3}$$

证毕.

引理 1. 如果两个浮点数 F_1、F_2 位于同一个区间,且尾数部分的前 n 位相同,则平均的 $LZC(F_1\oplus F_2)$ 等于 $W+2+n-(1/2^{t-n})$.

证明. 尾数部分共 t 位,如前 n 位相同,则变化的只可能是后 $t-n$ 位.将 $t-n$ 代入定理(1)中 t,再加上尾数部分相同的前 n 位,从而有

$$\overline{LZC(T_{F_1}\oplus T_{F_2})}=n+\frac{2^{t-n}-1}{2^{t-n}}=n+1-\frac{1}{2^{t-n}}\tag{4}$$

故 $\overline{LZC(F_1\oplus F_2)}=W+2+n-(1/2^{t-n})$.得证.证毕.

定理 2. 如果两个浮点数 F_1、F_2 位于同一个区间,且尾数部分的前 n 位相同,则它们之间的最大相对误差为 $1/2^n-1/2^t$.

证明. 若尾数部分前 n 位相同,相对误差最大的情况只能出现在尾数部分前 n 位均为 0,且其中一个浮点数(F_1/F_2)的后 $t-n$ 位为 0,而另一个浮点数(F_2/F_1)的后 $t-n$ 位为 1 的时候.这时两者之差的绝对值为

$$|F_1-F_2|=2^{E-2^{W-1}+1}(1/2^n-1/2^t)\tag{5}$$

两者绝对值的最小值为

$$\min(|F_1|,|F_2|)=2^{E-2^{W-1}+1} \tag{6}$$

式(5)除以式(6)之后可得两者之间的最大相对误差为 $1/2^n-1/2^t$. 得证. 　　证毕.

定理 3. 　如果两个浮点数 F_1、F_2 不位于同一个区间，但它们的前 n 位相同，则平均的 $LZC(F_1\oplus F_2)$ 约等于 $n+1$.

证明. 　类似引理 1，将符号位、指数部分和尾数部分共 $W+t+1$ 位作为一个整体考虑，将 $W+t+1$ 类比于引理 1 中的 t，将 n 类比于引理 1 中的 n，通过类似于定理 1 和引理 1 的证明可知：

$$\overline{LZC(F_1\oplus F_2)}=n+1-(1/2^{t+W+1-n})\approx n+1 \tag{7}$$

证毕.

定理 4. 　如果两个浮点数 F_1、F_2 不位于同一个区间，但它们的前 n 位相同，则它们之间的最大相对误差为 $2^{2^{W-n+1}}-1$.

证明. 　因 F_1、F_2 之间的最大相对误差为

$$\frac{|F_1-F_2|}{\min(|F_1|,|F_2|)}=\frac{|2^{E_1}(2^t+T_1)-2^{E_2}(2^t+T_2)|}{\min(|2^{E_1}(2^t+T_1)|,|2^{E_2}(2^t+T_2)|)} \tag{8}$$

最大相对误差当两个浮点数其中一个的后 $W-n+1$ 位指数位及尾数位全为 0，且另一个的后 $W-n+1$ 位指数位及尾数位全为 1 时取到. 此时 E_1 与 E_2 两者分别为 2^W-2^{W-n+1} 与 2^W-1，而 T_1 与 T_2 分别为 0 与 2^t-1. 代入式(8)，有最大相对误差为

$$\frac{|F_1-F_2|}{\min(|F_1|,|F_2|)}\approx\frac{2\times 2^{2^{W-1}}-2^{2^W-2^{W-n+1}}}{2^{2^W-2^{W-n+1}}}=2^{2^{W-n+1}}-1 \tag{9}$$

证毕.

但我们注意到，定理 2 及定理 4 的逆命题均不成立. 即，根据最大相对误差，并不能确保两个浮点数的二进制表示的前若干位相同. 为了建立平均相对误差到平均的 $LZC(F_1\oplus F_2)$ 的映射关系，有如下定理.

定理 5. 　如果两个浮点数 F_1、F_2 的符号位相同，且两者中绝对值较大者与绝对值较小者的比值为 r，则它们的区间距离为 $\lfloor\log_2 r\rfloor$ 或 $\lceil\log_2 r\rceil$.

证明. 　(1) 当 $\log_2 r$ 为整数时，这时因为 r 为 2 的整数次幂，故 F_1 与 F_2 的尾数部分相同. 根据式(1)，知此时的比值 r 为 $2^{|E_1-E_2|}$. 再根据定义 1，它们的区间距离为 $|E_{F_1}-E_{F_2}|$，也即 $\log_2 r$. 符合定理 5. (2) 当 $\log_2 r$ 为非整数时，此时记 F_1 与 F_2 中绝对值最小者为 $F_{\min}$，绝对值较大者为 $F_{\max}$. 假设 $F_{\min}$ 位于某一区间 i，则当 $|F_{\min}|$ 取最小值时，因 $|F_{\max}|>|F_{\min}|\times 2^{\lfloor\log_2 r\rfloor}$，且 $|F_{\max}|<|F_{\min}|\times 2^{\lceil\log_2 r\rceil}$，故此时 $F_{\max}$ 必位于区间 $i+(-1)^S\lfloor\log_2 r\rfloor$. 当 $|F_{\min}|$ 取最大值时，同样因 $|F_{\max}|>|F_{\min}|\times 2^{\lfloor\log_2 r\rfloor}$，且 $|F_{\max}|<|F_{\min}|\times 2^{\lceil\log_2 r\rceil}$，故此时 $F_{\max}$ 必位于区间 $i+(-1)^S\lceil\log_2 r\rceil$. 且因为区间 $i+(-1)^S\lfloor\log_2 r\rfloor$ 与区间 $i+(-1)^S\lceil\log_2 r\rceil$ 为相邻区间，故当 $F_{\min}$ 值在区间 i 内变化时，$F_{\max}$ 必定也在区间 $i+(-1)^S\lfloor\log_2 r\rfloor$ 与区间 $i+(-1)^S\lceil\log_2 r\rceil$ 变化. 也即两者的区间距离为 $\lfloor\log_2 r\rfloor$ 或 $\lceil\log_2 r\rceil$. 得证. 　　证毕.

引理 2. 　如果两个浮点数 F_1、F_2 的符号位相同，且两者中绝对值较大者与绝对值较小者的比值为 r，记 F_1 与 F_2 中绝对值较小者为 $F_{\min}$，$F_{\min}$ 所在区间的绝对值最小值的浮点数为 B. 则当 $F_{\min}/B\in[1,2^{\lceil\log_2 r\rceil}/r)$ 时，F_1 与 F_2 的区间距离为 $\lfloor\log_2 r\rfloor$，当 $F_{\min}/B\in[2^{\lceil\log_2 r\rceil}/r,2)$ 时，两者的区间距离为 $\lceil\log_2 r\rceil$.

证明. 　当 $F_{\min}/B\in[1,2^{\lceil\log_2 r\rceil}/r)$ 时，这时 $F_{\max}=F_{\min}\times r\in[rB,2^{\lceil\log_2 r\rceil}B)$，与 B 所在区间(也即 $F_{\min}$ 所在区间)区间距离为 $\lfloor\log_2 r\rfloor$. 当 $F_{\min}/B\in[2^{\lceil\log_2 r\rceil}/r,2)$ 时，这时 $F_{\max}=F_{\min}\times r\in[2^{\lceil\log_2 r\rceil}B,2rB)$，与 B 所在区间(也即 $F_{\min}$ 所在区间)区间距离为 $\lceil\log_2 r\rceil$. 得证. 　　证毕.

引理 3. 　如果两个浮点数 F_1、F_2 的符号位相同，且两者中绝对值较大者与绝对值较小者的比值为 r，则两者之间的平均区间距离为 $\lceil\log_2 r\rceil+1-2^{\lceil\log_2 r\rceil}/r$.

证明. 　根据引理 2，F_1 与 F_2 之间的平均区间距离应为

$$\begin{aligned}&(2^{\lceil\log_2 r\rceil}/r-1)\times\lfloor\log_2 r\rfloor+(2-2^{\lceil\log_2 r\rceil}/r)\times\lceil\log_2 r\rceil\\&=(2^{\lceil\log_2 r\rceil}/r-1)\times(\lceil\log_2 r\rceil-1)+\\&\quad(2-2^{\lceil\log_2 r\rceil}/r)\times\lceil\log_2 r\rceil\\&=\lceil\log_2 r\rceil+1-2^{\lceil\log_2 r\rceil}/r\end{aligned} \tag{10}$$

注意，本引理及上述证明对 r 为 2 的整数次幂亦成立. 得证. 　　证毕.

定义 3. 　区间内距离. 若浮点数 F_1 与 F_2 位于同一区间，定义 $|T_{F_1}-T_{F_2}|$ 为两浮点数之间的区间内距离. 可见，区间内距离的取值范围为 $[0,2^t-1]$.

定理 6. 　如果两个浮点数 F_1、F_2 位于同一区间，且区间内距离固定为 d，记 $\max(X,0)$ 为 $X^{\#}$，则它们之间的平均 $LZC(T_{F_1}\oplus T_{F_1})$ 为 $\sum_{i=1}^{t}(2^t-2^i d)^{\#}/(2^t-d)$.

证明. 　当区间内距离为 d 时，记 F_1 与 F_2 中较大者为 $F_{\max}$，较小者为 $F_{\min}$，则二元对 $(F_{\min},F_{\max})$ 共

有 2^t-d 种不同组合. 假设这 2^t-d 种组合均匀分布,平均每种出现一次,这时容易验证当 $t=1$ 时,定理 6 成立,即平均的 $LZC(T_{F_1}\oplus T_{F_1})$ 为 $(2-2d)^{\#}/(2-d)$. 设定理 6 对 $t=j$ 时成立,则这时 2^t-d 种组合每种出现一次时总的 $LZC(T_{F_1}\oplus T_{F_1})$ 为 $\sum_{i=1}^{j}(2^j-2^i d)^{\#}$. 那么当 $t=j+1$ 时,因为这时尾数 T 的 2^j+1 种组合可以看成是在 $t=j$ 的 2^j 种组合的基础上,前面分别补上了 0 和 1,且在最前面补多了一位,除了 $F_{\min}$ 尾数首位为 0 且 $F_{\max}$ 尾数首位为 1 的 d 种组合外,剩下的 $((2^{j+1}-d)-d)^{\#}=(2^{j+1}-2d)^{\#}$ 种组合的 $LZC(T_{F_1}\oplus T_{F_1})$ 均增加 1. 从而有如下递推关系:

$$
\begin{aligned}
&\sum_{(F_{\min},F_{\max})} LZC(T_{F_1}\oplus T_{F_2})|_{t=j+1}\\
&=\sum_{(F_{\min},F_{\max})} LZC(T_{F_1}\oplus T_{F_2})|_{t=j}\times 2+(2^{j+1}-2d)^{\#}\\
&=\Big[\sum_{i=1}^{j}(2^j-2^i d)^{\#}\Big]\times 2+(2^{j+1}-2d)^{\#}\\
&=\Big[\sum_{i=1}^{j}(2^{j+1}-2^{i+1} d)^{\#}\Big]+(2^{j+1}-2d)^{\#}\\
&=\sum_{i=0}^{j}(2^{j+1}-2^{i+1} d)^{\#}=\sum_{i=1}^{j+1}(2^{j+1}-2^{i} d)^{\#} \qquad (11)
\end{aligned}
$$

即当 $t=j+1$ 时,$2^{j+1}-d$ 种组合每种出现一次时总的 $LZC(T_{F_1}\oplus T_{F_1})$ 为 $\sum_{i=1}^{j+1}(2^{j+1}-2^i d)^{\#}$. 此时平均的 $LZC(T_{F_1}\oplus T_{F_1})$ 为 $\sum_{i=1}^{j+1}(2^{j+1}-2^i d)^{\#}/(2^{j+1}-d)$. 即定理 6 对 $t=j+1$ 亦成立. 由归纳法知,当 t 为浮点数的尾数部分位数时,定理 6 亦成立. 得证. 证毕.

3.3 高性能无损压缩方法

算法能得到的压缩比与预测误差直接相关,其关键在于预测误差前导零 $LZC(F_1\oplus F_2)$ 的平均个数. 因此,为了得到高性能无损压缩方法,本文继续针对 F_1、F_2 位于同一区间及不同区间的情况下,$LZC(F_1\oplus F_2)$ 的平均值进行理论分析.

引理 4. 如果两个浮点数 F_1、F_2 位于同一区间,且区间内距离固定为 d,则

(1) 它们之间的平均 $LZC(T_{F_1}\oplus T_{F_1})$ 当 $d=0$ 时为 t,当 $d\neq 0$ 且 $d\le 2^{t-1}$ 时可近似为 $[(t-2-\log_2 d)2^t+2d]/(2^t-d)$,当 $d>2^{t-1}$ 时为 0;

(2) 它们之间的平均 $LZC(F_1\oplus F_2)$ 当 $d=0$ 时为 $W+1+t$,当 $d\neq 0$ 且 $d\le 2^{t-1}$ 时可近似为 $W+1+[(t-2-\log_2 d)2^t+2d]/(2^t-d)$,当 $d>2^{t-1}$ 时为 $W+1$. 最坏情况下的近似相对误差约为 1%.

证明.

命题(1). 容易验证当 $d=0$ 时平均 $LZC(T_{F_1}\oplus T_{F_1})$ 为 t. 当 d 为 2 的整数次幂时,定理 6 的结果等于 $[(t-2-\log_2 d)2^t+2d]/(2^t-d)$,当 $d>2^{t-1}$ 时为 0. 当 d 不是 2 的整数次幂时,设 $d\in[2^j+1,2^{j+1}]$,此时根据定理 6 算出的准确结果为 $[(t-j-1)2^t-2^{t-j}d+2d]/(2^t-d)$,与引理 4 的近似值之间的误差为 $[\log_2 d-(j-1+2^{-j}d)]\,2^t/(2^t-d)$. 设 $j_1=\log_2 d-j$,其中 $j_1\in(0,1]$,故近似误差又可表示为 $(1+j_1-2^{j_1})\,2^t/(2^t-d)$. 易知,当 $j=t-2$,$j_1\approx 0.58$ 的时候,这时的近似误差取得全局最大值(最坏情况)约为 0.136. 命题(1)得证.

命题(2). 因为最后压缩的时候,是对整个浮点数进行编码,故需要考虑整个预测误差的前导零数,即 $LZC(F_1\oplus F_2)$. 根据定义 1 和定义 2,当 F_1、F_2 位于同一区间时,$LZC(F_1\oplus F_2)=W+1+LZC(T_{F_1}\oplus T_{F_1})$,故命题(2)的前 3 部分得证. 根据定理 6,对于 64 位浮点数而言,这时的 $LZC(F_1\oplus F_2)$ 的准确值为 14. 故最坏情况下的近似相对误差约为 $0.136/14\approx 1\%$. 得证. 证毕.

引理 5. 如果两个浮点数 F_1、F_2 位于不同区间,且它们的区间距离为 $d(d>0,d\le 2^W)$,则它们之间的平均 $LZC(F_1\oplus F_2)$ 可近似为 $[(W-1-\log_2 d)2^{W+1}+2d]/(2^{W+1}-d)$,近似的平均相对误差约为 1%~2%.

证明. 将符号部分与指数部分共 $W+1$ 位,类比于引理 4 中的尾数部分 t 位,除物理意义不同外,所有的推算不变. 根据引理 4 的命题 1,即得引理 5 的近似值. 同样根据引理 4 可知若 $d\in[2^j+1,2^{j+1}]$,准确结果为 $[(W-j)2^{W+1}-2^{W+1-j}d+2d]/(2^{W+1}-d)$,近似的相对误差为 $[(j-1+2^{-j}d)-\log_2 d]2^{W+1}/[(W-j)2^{W+1}-2^{W+1-j}d+2d]$. 因这时的区间距离为指数部分之差,即使实际值与预测值之间差 10^{30} 倍,它们之间的区间距离 d 也只是为 $100(2^{100}\approx 10^{30})$. 也就是说,$d$ 的取值通常较小. 为了对近似所引起的误差有直观认识,分别取 $d=\{1,2,\cdots,100\}$,图 3 表述了准确值及近似值随 d 变化的关系. 当 $d\in[1,100]$ 时,计算可知平均相对误差约为 1.4%,得证. 证毕.

根据引理 2 可知,若两个浮点数 F_1、F_2 的符号位相同,且两者中绝对值较大者与绝对值较小者的比值为 $r(r\ge 1)$,则当 $r<2$ 时,F_1 与 F_2 有可能位于同一区间(概率为 $2/r-1$)或者相邻区间(概率为

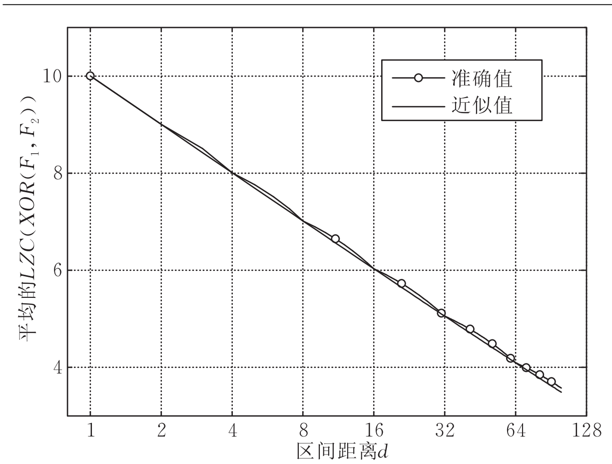

图 3　$LZC(F_1\oplus F_2)$的近似值和准确值之间的误差

$2-2/r$)；当 $r\geq 2$ 时，F_1 与 F_2 则不可能位于同一区间.

$r<2$ 的情况. 此时，若 F_1 与 F_2 位于同一区间，设所在区间的绝对值最小值的浮点数为 B，此时 F_1 的范围为$[B,2B/r)$，区间内距离 $d\in[(r-1)\times 2^t,(2-2/r)\times 2^t)$. 当 $r\in(1,4/3]$时，$(r-1)\times 2^t\leq 2^{t-1}$ 且$(2-2/r)\times 2^t\leq 2^{t-1}$；当$r\in(4/3,3/2]$时，$(r-1)\times 2^t\leq 2^{t-1}$ 且$(2-2/r)\times 2^t>2^{t-1}$；当 $r\in(3/2,2)$时，$(r-1)\times 2^t>2^{t-1}$ 且$(2-2/r)\times 2^t>2^{t-1}$. 设引理 4 中的 $W+1+[(t-2-\log_2 d)2^t+2d]/(2^t-d)$ 为 $f(d)$，因为

$$\int f(x)\mathrm{d}x=(W-1)x-t2^t\ln(2^t-x)+\frac{2^t}{\ln 2}\left[Li_2\left(1-\frac{x}{2^t}\right)+\ln(x)\ln\left(1-\frac{x}{2^t}\right)\right]\tag{12}$$

其中 $Li_2(*)$为二重对数(dilogarithm)函数. 所以，当 F_1 与 F_2 位于同一区间，$\overline{LZC(F_1\oplus F_2)}_{同}$ 为

$$\frac{1}{\left(3-r-\frac{2}{r}\right)\times 2^t}\times\begin{cases}\displaystyle\int_{(r-1)\times 2^t}^{(2-2/r)\times 2^t} f(x)\mathrm{d}x, & r\in(1,4/3]\\ \displaystyle\left(\frac{3}{2}-\frac{2}{r}\right)2^t(W+1)+\int_{(r-1)\times 2^t}^{2^{t-1}} f(x)\mathrm{d}x, & r\in\left(\frac{4}{3},\frac{3}{2}\right]\\ W+1, & r\in(3/2,2)\end{cases}\tag{13}$$

将式(12)代入式(13)，并进行化简后可得当$r\in(1,4/3]$时，式(13)成为

$$W-1+\frac{\left[Li_2(r-1)-Li_2\left(2-\frac{2}{r}\right)\right]r}{(3r-r^2-2)\ln 2}\tag{14}$$

当 $r\in(4/3,3/2]$时，式(13)成为

$$W-1+\frac{\left(3-\frac{4}{r}\right)\ln 2-Li_2\left(\frac{1}{2}\right)+Li_2(r-1)}{(3-r-2/r)\ln 2}\tag{15}$$

再根据引理 5，$\overline{LZC(F_1\oplus F_2)}_{异}$ 为$[(W-1)2^{W+1}+2]/(2^{W+1}-1)$，再根据 F_1 与 F_2 有 $2/r-1$ 的概率同区间，有 $2-2/r$ 的概率位于相邻区间，故

$$\overline{LZC(F_1\oplus F_2)}=\overline{LZC(F_1\oplus F_2)}_{同}\times(2/r-1)+\overline{LZC(F_1\oplus F_2)}_{异}\times(2-2/r)\tag{16}$$

$r\geq 2$ 的情况. 此时，根据引理 2 可知有 $2^{\lceil\log_2 r\rceil}/r-1$的概率 F_1 与 F_2 的区间距离为$\lfloor\log_2 r\rfloor$，$2-2^{\lceil\log_2 r\rceil}/r$ 的概率两者的区间距离为$\lceil\log_2 r\rceil$. 此时的平均区间距离可近似为 $\log_2 r$，再根据引理 5，可知此时的平均 $LZC(F_1\oplus F_2)$为

$$\frac{(W-1-\log_2(\log_2 r))2^{W+1}+2\log_2 r}{2^{W+1}-\log_2 r}\tag{17}$$

设原始浮点数及预测器的预测值分别为 F 与 $\tilde{F}$，当两者符号位相同时，此时的最大预测误差 err 为 $r-1$($err\geq\tilde{F}/F-1$). 而因为 F 与 $\tilde{F}$ 共有 $LZC(F\oplus\tilde{F})$个前导零，假设极限情况时冗余前导零的信息量趋近零，则此时的压缩比 R_c为$(W+1+t)/[W+1+t-LZC(F\oplus\tilde{F})]$. 根据式(13)～(17)，图 4 描述了最大预测误差 err 与压缩比 R_c之间的关系. 此关系仅适用于 F 与 $\tilde{F}$ 符号位相同的情况，当符号位不同时，因此时的 $LZC(F\oplus\tilde{F})$为 0，故采用其他方法进行压缩(见第 3.3.2 与 3.3.3 节).

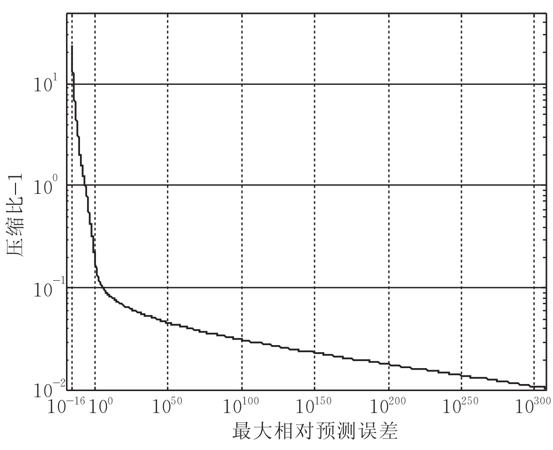

图 4　预测误差与压缩比之间的关系

从图 4 中可见，预测的准确性直接影响到压缩比. 当相对预测误差小于 1 时，压缩比以较快的下降速度从 25 下降到 1.2 左右. 当相对预测误差从 1 增加到 10^{300}时，压缩比也缓慢地由 1.2 降低到 1.01. 分析表明，为了得到较高的压缩比，就必须使用较准

确的预测算法,或者混合使用多种预测算法.

进行高性能科学数据压缩的关键在于根据浮点数的存储格式,设计先进的预测算法,并对原始值 F 与预测值 $\widetilde{F}$ 之间的差值进行高性能编码. 本文所提出的科学数据高性能无损压缩系统的结构如图 5 所示.

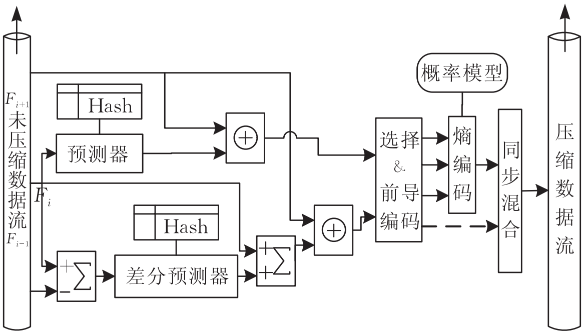

图 5 编码器结构图

系统结果主要分为预测和编码两部分,将在以下章节中对其进行详细分析.

3.3.1 数据预测

根据图 4,数据预测的准确性直接关系到最终的压缩比. 因科学数据的不同,使用单一的预测算法往往不能永远得到很好的预测精度,而预测精度的下降将会导致压缩比的急剧下降. 为此,有必要使用混合式预测算法,同时采用多种预测器,然后再在多个预测结果中选择最接近真实值的预测结果.

但采用多种预测算法也会带来一定的负面影响,首先,会增加编码器和解码器的复杂性. 虽然在多处理器系统上或者使用硬件实现时,多个预测器可以并发工作,但对多个预测器进行同步仍会影响到系统的效率. 而在串行系统上,预测部分的执行时间会随着预测器数目的增多而线性增加. 其次,预测之后需要记录预测器的编号以用于解码,从而会增加需要编码的数据量.

为了在预测精度和速度之间取得一个平衡,本文使用 2 个预测器,一个用于预测比较平缓的数据,另一个差分预测器用于预测变化的数据. 如图 5 所示,普通预测器以 F_i 的值作为索引,查找预测表,得出 F_{i+1} 的预测值 $\widetilde{F}_{i+1}$, $\widetilde{F}_{i+1}$ 与 F_{i+1} 的异或值将会送给选择器和前导编码器进行编码. 对于差分预测器而言,区别在于预测器是根据 F_i-F_{i-1} 来得到 $F_{i+1}-F_i$ 的预测值. 所以差分预测器的输入是 F_i-F_{i-1},而其输出与 F_i 求和之后作为 F_{i+1} 的预测值 $\widetilde{F}_{i+1}$. 差分预测器的工作机理决定了差分预测器能较好地预测线性增长的数据(一阶差). 因普通浮点数的加减法有时是不可逆的,因此图 5 中的求和器使用的是与浮点数二进制相对应的整数的加减法. 本文采用的普通预测器为 FCM[14],采用的差分预测器为 DFCM[15]. 实验表明,这两种预测器对于浮点数据能有较好的预测精度. 为了减少预测表的存储空间,本文采用 Hash 表的方式来存储预测表. 而在实现时,本文使用前面若干个真实值的组合作为 Hash 表的索引,来得到预测值,从而增加预测的精度[8].

3.3.2 异或编码和前导编码

在得到真实值 F 的预测值 $\widetilde{F}$ 之后,本文将使用熵编码方法来编码 F 与 $\widetilde{F}$ 的"差值"ΔF. 具体来说,本文将 ΔF 定义为 $F\oplus\widetilde{F}$,并将 ΔF 分为前导零部分和非零部分. 对于前导零部分将使用熵编码进行压缩传输,而对于非零部分将进行非压缩传输. 设 ΔF 的符号位为 S,设除符号位外,ΔF 的最后非零位在第 k 位($k\in[0,W+t-1]$),则当 ΔF 非零时,使用熵编码传输 S 与 k,再无压缩传输的后 k 位(当 $k=0$ 时,无需传输非零部分). 当 ΔF 为零时,则只需要使用熵编码传输 0. 本文的 ΔF 定义与文献[7]的定义相同,但不同于文献[7]直接传输 ΔF,本文根据浮点数的存储结构,将 ΔF 分为符号位和非符号位分别处理. 对于符号位和前导零,使用快速的变长熵编码压缩方式,从而可以获得更高的压缩比. 比如:若 F 与 $\widetilde{F}$ 分别为 -0.01 和 0.011,这时的 ΔF 所对应的 16 进制表示为 8002FDCA45A25DC1. 若使用本文的方法,除了前导零熵编码之外,只需要存储后面的 49 位,而文献[7]的方法则需要存储所有的 64 位,还需要 4 位来定长编码前导零. 文献[4]将 ΔF 定义为 $F-\widetilde{F}$,虽然这种定义更为直接,主要的缺点包括:

(1) 因直接对浮点数的减法有时并不可逆,即使用 $F-\widetilde{F}$ 得到 ΔF,但用 $\widetilde{F}+\Delta F$ 并不一定能准确无损地得到 F. 特别是当编码器和解码器位于不同的系统,使用了不同的编译选项时更是如此. 然而,对于现实中的科学数据压缩,不可能要求编码和解码都在同一系统上,使用同一程序完成. 所以,在对 F 与 $\widetilde{F}$ 进行减法操作前,就必须将它们映射成整数进行操作. 若采用常规的映射方法,则最小正浮点数将会映射成 00…0,而最大负浮点数将会映射成 10…0,从而使得两者的整数距离很大. 为解决这个问题,文献[3-4]都采用特别的映射方法. 这种额外的映射将会增加编码器的复杂度,并降低编解码的效率.

(2) 需要记录 ΔF 的符号位信息. 因 ΔF 大于零及小于零的概率基本相等,所以即使使用熵编码,也需要约 1 比特来存储符号位信息.

总体而言，实验表明，两种 ΔF 定义所获得的压缩比没有明显区别，但本文所采用的方式在编解码速度上更为高效.

3.3.3　高效熵编码

通过图 5 所示的“选择与前导编码器”之后，输出数据包括预测器标志位 C（1 位），ΔF 的符号位 S（1 位）以及除符号位外，ΔF 的最后非零位位置 k. k 的范围为 $[0, W+t-1]$. 本文使用 $k=W+t$ 来特别标记除符号位外，ΔF 的其他位全零的情况. 故熵编码的输入为三元组 $\langle C, S, k\rangle$. 本文使用 $(k \ll 2 + C \ll 1 + S)$ 的移位运算将三元组转换成整数，再使用熵编码器对该整数进行编码.

本文选择 Range 编码[16] 作为熵编码方法. Range 编码的机理与算法编码类似，但与算术编码的输出以位为单位不同，Range 编码的输出以字节为单位. 所以，在现代计算机系统结构下，Range 编码的速度会比算法编码要快，且压缩比类似. 同时，Range 编码不受专利的限制，可以自由使用在科学研究系统中.

本文选择准静态概率模型（Quasistatic Probability Model）[17] 作为 Range 编码器的概率模型. 准静态概率模型并不在编码每个符号后都更新概率表，而只在每编码若干个符号之后，才更新一次概率表. 通过准静态概率模型和 Range 编码器，本系统的熵编码部分能在编码效率和压缩比之间取得良好的平衡.

熵编码后的三元组与 ΔF 的后 k（k 可能为 0）位通过同步混合，从而形成了最终的压缩数据流.

3.3.4　解码器

与编码器相对应的解码器结构如图 6 所示. 普通预测器根据 F_i 得到预测值 $\widetilde{F}_{i+1,1}$，差分预测器根据 F_i 和 F_{i-1} 得到预测值 $\widetilde{F}_{i+1,2}$. 压缩数据流通过熵解码，得到三元组 $\langle C, S, k\rangle$. 前导解码器根据 S 和 k 进行前导解码，得到 ΔF. “2 选 1”处理根据 C 来决定是用 $\widetilde{F}_{i+1,1}$ 还是 $\widetilde{F}_{i+1,2}$ 来作为 $\widetilde{F}_{i+1}$. 最后，通过异或 $\widetilde{F}_{i+1}$ 与 ΔF 得到 F_{i+1}.

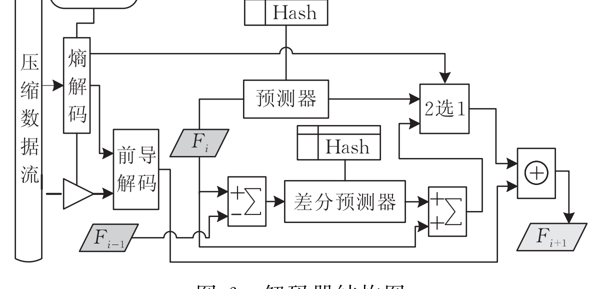

图 6　解码器结构图

4　性能评估及推广应用

4.1　性能评估与分析

下面通过将本文提出的科学数据无损压缩方法与其他的压缩方法在压缩速度和压缩比两方面进行比较，从而说明本文提出的科学数据无损压缩方法在速度和效果两方面的高性能.

4.1.1　评价指标

主要的评价指标包括：

(1) 压缩比. 压缩之前数据量与压缩之后的数据量之间的比值. 为了减少极大值的影响，本文采用几何平均值作为多个数据压缩比的平均值.

(2) 吞吐量/速度. 单位时间内能处理的数据量的大小. 分为压缩吞吐量和解压缩吞吐量，以 Mb/s 或 MB/s 为单位（1MB/s＝8Mb/s）. 运算吞吐量与运算时间成反比，若压缩吞吐量为 X MB/s，则意味着每压缩 1MB 的数据需要 $1/X$ 秒.

(3) 内存使用量. 压缩比与内存使用量之间的关系.

4.1.2　实验环境

为了说明本压缩算法的适用性，实验在一台普通的笔记本电脑上进行. 处理器为 1.83G 的 32 位 Intel 双核 Core Duo T2400. 为了减小并行双核的影响，在 BIOS 中禁用了双核功能. 该处理器的 L2 缓存为 2MB，前端总线为 667MHz. 系统内存为双通道 DDR2 667MHz 2GB（1GB × 2）. 操作系统为 Ubuntu Linux 8.04（Hardy），内核版本为 2.6.24-24. 程序采用 gcc 4.2.4（g++）编译，编译选项只简单的使用了”－O3”选项.

4.1.3　实验设计

为了与其他压缩算法具有对比性，本文没有刻意对具体实现进行优化. 所有的输出均重定向到了/dev/null. 测试数据集采用文献[8]的科学数据. 参与比较的压缩算法和软件包括 Bzip2、FPC[8-9]、Gzip 和 PLMI[4]. 为了使所有的输入文件都缓存到内存，所有的程序都执行了 5 遍，以最短的时间作为执行时间. 所有的时间值以系统的 time 命令所得出的用户态和内核态（系统态）时间之和为准. 内存使用量以系统报告的为准.

4.1.4　分析与评价

通过对各种算法分析比较，得出了压缩比比较表，如表 1 所示. 其中 Bzip2 和 Gzip 均采用 1～9 压缩等级中最好的结果. FPC 采用 1～25 压缩等级中最好

的结果.各种数据上最优的压缩结果采用粗体标出.

表1　各种算法的压缩比比较

	Bzip2	FPC	Gzip	PLMI	本文
msg_bt	1.102	1.288	1.130	1.199	**1.374**
msg_lu	1.021	1.173	1.055	1.134	**1.266**
msg_sppm	6.933	5.298	7.431	3.249	**7.760**
msg_sp	1.075	1.262	1.108	1.112	**1.360**
msg_sweep3d	1.294	3.089	1.092	1.330	**4.036**
num_brain	1.043	1.164	1.064	**1.245**	1.244
num_comet	1.173	1.157	1.162	**1.265**	1.249
num_control	1.030	1.050	1.058	1.124	**1.147**
num_plasma	5.789	15.048	1.608	1.063	**227.0**
obs_error	1.339	3.603	1.448	1.365	**4.421**
obs_info	1.217	2.273	1.154	1.056	**2.544**
obs_spitzer	**1.752**	1.027	1.232	1.075	1.101
obs_temp	1.024	1.019	1.036	1.088	**1.100**
算术平均	1.984	2.958	1.660	1.331	**19.662**
几何平均	1.520	1.945	1.347	1.263	**2.702**

由表1可看出,本文提出的方法在大多数数据集上明显优于其他压缩算法.因为"num_plasma"数据的重复率比较大,且变化较缓慢,从而预测精度较高,预测误差前导零的个数多,熵编码效率高,所以得到了高达227的压缩比.所得到的算术平均压缩比和几何平均压缩比也明显高于其他算法.为了减少极大值对平均值的影响,以下均以几何平均值为准.

图7描述各种方法及其相应的各种压缩等级与几何平均压缩比之间的关系.

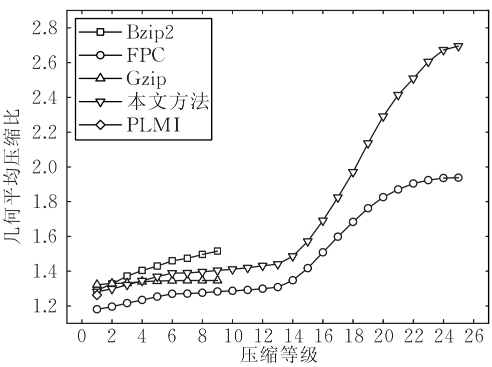

图7　各种方法及其相应的各种压缩等级与几何平均压缩比之间的关系

从图7中可看到,多数压缩算法的平均压缩比只能达到1.5,FPC的平均压缩比最高可达1.94(压缩等级为25),此时FPC算法需要514MB的内存空间(图8).而本文算法的平均压缩比最高可达2.7(压缩等级为25),远高于其他算法.当压缩等级为18时即可得到1.97的平均压缩比(图7),此时本文算法所需要的内存空间仅为7MB(图8).所以,不难看出,不管是在最重要的压缩比上,还是空间复杂度上,本文的算法均要远优于其他算法.

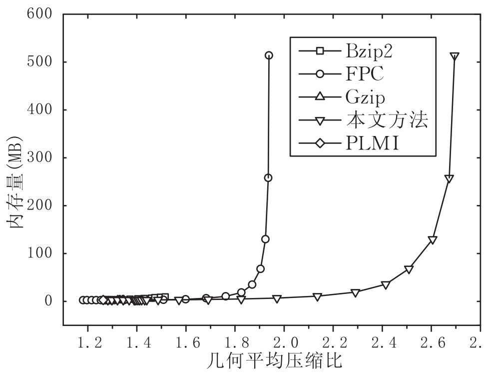

图8　各种方法的平均压缩比与内存使用量之间的关系

图9描述了压缩比与系统压缩吞吐量之间的关系.当使用"cat file>/dev/null"实测不压缩时(压缩比为1),系统的最大吞吐量为12Gb/s.故所有的吞吐量均根据系统最大吞吐量进行了均一化.从图中可看到,所有其他算法的吞吐量都随着压缩比的增加而急剧减少,而本文方法的吞吐量随着压缩比的变化基本保持不变.FPC方法在低压缩比时拥有较高的吞吐量.但当压缩比较高时(>1.8),FPC算法的吞吐量与本文相近.而在此时,FPC算法需要的内存远高于本文方法.

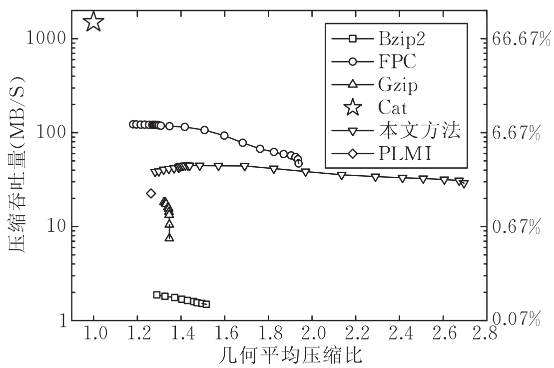

图9　各种方法的平均压缩比与压缩吞吐量之间的关系

以上的实验结果及分析表明:

(1)本文所提出的科学数据无损压缩方法的压缩比均远优于其他算法.本文方法的最高平均压缩比可高达2.7,而其他算法的最高平均压缩比只能达到1.94.

(2)本文方法的空间复杂度(内存使用量)同样优于其他算法.在低压缩比(1→1.7)时,各种算法的内存使用量相近,均为2MB左右.而当中等压缩比(1.7→2)时,本文方法的内存使用量远低于其他算法.只有本文方法能达到高压缩比(>2).

(3)本文方法的压缩吞吐量随着压缩比的变化基本保持恒定.在低压缩比时,系统吞吐量优于Gzip,远优于Bzip2,但低于FPC.当中等和较高压

缩比时，吞吐量与 FPC 相近.

4.2 压缩其他格式的浮点数数据

虽然之前所描述的算法以及性能评估都是以 64 位浮点数为例，但本科学数据无损压缩方法同样适用于其它格式的浮点数数据，比如 32 位的浮点数和 128 位的浮点数数据压缩.

数据的压缩比与科学数据的本身精度相关. 64 位浮点数能表示的范围比 32 位浮点数更宽，且精度更高. 对于同一个应用而言，相比 32 位浮点数而言，当使用 64 位浮点数存储时，对存储位的利用率更高. 如当存储圆周率 PI 时，若使用 32 位浮点数，需要用到 1 个符号位、1 个指数位和 23 个小数位（总体利用率 78%）；而若使用 64 位浮点数，需要用到1 个符号位、1 个指数位和 52 个小数位（总体利用率 84%）. 所以，对同一个应用，32 位数据比 64 位数据稍易进行压缩.

根据式(14)，也可得到类似的结论. 当预测误差为 1%(r=1.01)时，若使用的是 32 位浮点数，根据式(14)知可压缩的前导零数目约为 13.2，压缩比约为 1.7. 而若使用的是 64 位浮点数，同样根据式(14)知可压缩的前导零数目约为 16.2，压缩比约为 1.34.

4.3 应 用

本文方法主要应用于科学计算中二进制原始浮点数据的无损压缩. 因压缩后的数据小于原始数据，采用压缩算法有利于节省存储空间及减少数据传输时间. 若用户需要将一个 1GB 的数据通过网络从一个地点传输到另一地点，即使采用的是 100Mb/s 的高速以太网（实际速率约为 10MB/s），在最理想情况下需要的传输和存储时间为 100s，且接收者需要存储 1GB 的数据.

为了减少需要传输和存储的数据量，许多科学计算工作者采用通用压缩工具（如 Gzip）压缩数据. 仍以上述分析为例，Gzip 使用默认参数时几何平均压缩比约为 1.34，压缩吞吐率约为 16MB/s，解压缩吞吐率约为 75MB/s. 考虑到压缩，解压缩和传输可由处理器和网络适配器同时进行，若在传输的同时进行压缩和解压缩，则总时间只由系统瓶颈决定. 由图 9 可知，除 Bzip2 外，各压缩算法的吞吐率均大于 10MB/s，大于网络传输速率. 在计算与传输重叠时，总时间约为(1000MB/1.34)/(10MB/s)≈74.6s；计算与传输串行时，总时间约为 74.6s＋1000MB/(16MB/s)＋1000MB/(75MB/s) ≈ 150s. 且只需要存储 746MB 的数据.

而若采用本文方法，当几何平均压缩比为 2.7 时，压缩吞吐率约为 29MB/s，解压缩吞吐率约为 25MB/s. 在计算与传输重叠时，总时间约为(1000MB/2.7)/(10MB/s)≈37s；计算与传输串行时，总时间约为 37s＋1000MB/(29MB/s)＋1000MB/(25MB/s) ≈ 111s. 且只需要存储 370MB 的数据. 由此可见，在计算与传输重叠时，通过压缩数据带来的性能提升是很明显的. 即使在计算与传输串行时，也能在速度相近的情况下，大幅度减少需要的存储空间.

而实际的互联网传输速率往往远低于100Mb/s，以 ADSL 宽带为例，最大上行速率 512kb/s，最大下行速率 8Mb/s(＝1MB/s)，导致传输时间大大增加，从而采用本文方法的效果更为明显. 本文方法已被应用于颗粒材料的大规模动力学过程仿真中. 主要包括颗粒材料的堆积过程模拟研究和颗粒的相分离现象模拟研究. 在这些仿真中，涉及到 10^7 规模的颗粒体在真三维环境中的动力学过程，每个模拟步产生数 GB 的数据. 为了对这些海量数据进行存储，必须使用本文的高性能无损压缩方法. 这些应用的具体信息请参见文献[18-21].

5 总结及未来工作

本文提出了高性能的科学数据无损压缩算法. 通过与其他算法进行比较，本文所提出的方法的压缩比均远优于其他算法，最高平均压缩比可高达 2.7，而其他算法的最高平均压缩比只能达到 1.94. 本文方法的空间复杂度（内存使用量）同样优于其他算法. 在低压缩比(1→1.7)时，各种算法的内存使用量相近，均为 2MB 左右. 而当中等压缩比(1.7→2)时，本文方法的内存使用量远低于其他算法. 同时，本文方法在低、中、高压缩比时，均具有高吞吐量的特点，数据处理能力在中、高压缩比时远高于通用压缩算法. 本文提出的方法特别适合海量科学数据的高压缩比无损传输和存储.

在本文中，只考虑了前导零的熵编码，在接下来的研究中将继续考虑后缀零的影响. 同时，因为预测算法和预测精度对于压缩比至为关键，在后续研究中还将考虑其他多种预测算法以及基于概率模型的预测算法自适应选择和混合. 在应用方面，除了已应用了本文方法的大规模颗粒动力学模拟之外，还将根据其他大规模实际科学计算和工程仿真的需求，对本文方法进行进一步推广和改进.

参 考 文 献

[1] Taubin G, Rossignac J. Geometric compression through topological surgery. ACM Transactions on Graphics, 1998, 17(2): 84-115

[2] Touma C, Gotsman C. Triangle mesh compression//Proceedings of the Graphics Interface'98. Vancouver, Canada, 1998: 26-34

[3] Isenburg M, Lindstrom P, Snoeyink J. Lossless compression of predicted floating-point geometry. Computer-Aided Design, 2005, 37(8): 869-877

[4] Lindstrom P, Isenburg M. Fast and efficient compression of floating-point data. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(5): 1245-1250

[5] Ibarria L, Lindstrom P, Rossignac J, Szymczak A. Out-of-core compression and decompression of large n-dimensional scalar fields//Computer Graphics Forum (Eurographics 2003). Granada, Spain, 2003: 343-348

[6] Engelson V, Fritzson D, Fritzson P. Lossless compression of high-volume numerical data from simulations//Proceedings of the 2000 Data Compression Conference. Snowbird, Utah, USA, 2000: 574-586

[7] Ratanaworabhan P, Ke J, Burtscher M. Fast lossless compression of scientific floating-point data//Proceedings of the 2006 Data Compression Conference. Snowbird, Utah, USA, 2006: 133-142

[8] Burtscher M, Ratanaworabhan P. High throughput compression of double-precision floating-point data//Proceedings of the 2007 Data Compression Conference. Snowbird, Utah, USA, 2007: 293-302

[9] Burtscher M, Ratanaworabhan P. FPC: A high-speed compressor for double-precision floating-point data. IEEE Transactions on Computers, 2009, 58(1): 18-31

[10] IEEE 754: Standard for binary floating-point arithmetic, 1985

[11] IEEE 754-2008: Standard for floating-point arithmetic, 2008

[12] Mahoney M. Adaptive weighing of context models for lossless data compression. Florida Institute of Technology, Technical Report TR-CS-2005-16, 2005

[13] Cleary J G, Witten I H. Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications, 1984, 32(4): 396-402

[14] Sazeides Y, Smith J E. The predictability of data values//Proceedings of the 30th International Symposium on Microarchitecture. Research Triangle Park, North Carolina, USA, 1997: 248-258

[15] Goeman B, Vandierendonck H, Bosschere K. Differential FCM: Increasing value prediction accuracy by improving table usage efficiency//Proceedings of the International Symposium on High Performance Computer Architecture. Nuevo Leone, Mexico, 2001: 207-216

[16] Martin G N N. Range encoding: An algorithm for removing redundancy from a digitized message//Video and Data Recording Conference. Southampton, England, 1979

[17] Schindler M. A fast renormalization for arithmetic coding//Proceedings of the 1998 Data Compression Conference. Snowbird, Utah, USA, 1998: 572

[18] He K J, Dong S B, Zhou Z Y. Multigrid contact detection method. Physical Review E, 2007, 75(3): 036710

[19] Zhong W Z, He K J, Zhou Z Y, Xia W, Li Y Y. Calibration of the damping coefficient for the DEM simulation. Acta Physica Sinica, 2009, 58(8): 5155-5161

[20] Zhong W Z, He K J, Zhou Z Y, Xia W, Li Y Y. Physical model and simulation system of powder packing. Acta Physica Sinica, 2009, 58(13): S21-S28

[21] Li Y Y, Xia W, Zhou Z Y, He K J, Zhong W Z, Wu Y B. Simulation of random mixed packing of different density particles. Chinese Physics B, 2010, 19(2): 024601

HE Ke-Jing, born in 1983, Ph. D., associate professor. His research interests include scientific computing, simulation, parallel computing and distributed systems.

Background

The research is supported by the National Natural Science Foundation of China (grant No. 10805019), the Natural Science Foundation of Guangdong Province (grant No. 8451064101000083), and the Fundamental Research Funds for the Central Universities, SCUT (grant No. 2009ZM0040).