УДК 004.043

Гамаюн В.П., д.т.н.

МЕТОДЫ СЖАТИЯ ДАННЫХ НА ОСНОВЕ КОДИРОВАНИЯ ЗНАЧАЩИХ РАЗРЯДОВ

Национальный авиационный университет

kafpi_itp@ukr.net

Предложены методы сжатия для двоичного представления данных в компьютерной среде на основе кодирования ненулевых, значащих разрядов в векторно-матричном блоке и линейном списке с использованием межразрядных расстояний – методом приращений

Ключевые слова: структуры данных, кодирование массивов данных

Введение и постановка задачи

Различные структуры данных, применяемые при обработке информации в ЭВМ можно представить как матрично-векторные блоки, элементами в которых есть „0" или „1". В такие структурные единицы данных возможно превратить последовательные файлы, векторные файлы, матричные файлы и другие, более сложные структуры данных. Одна из основных черт разрядно-логарифмического представления является кодирование только ненулевых разрядов – значащих единиц (ЗЕ).

Методы решения

Рассмотрим подход кодирования массивов данных в компьютерной среде, который предусматривает кодирование только значимых единиц, что возможно обеспечит меньший объем данных при хранении и передаче. Такой подход сжатия данных реализуется по следующему общему алгоритму[1, 2]. Структура данных, сжимается, переформатируется в матрично-векторной блок, который имеет описание по двум параметрам – количеству операндов или структурных единиц обработки (m) и разрядности структурных единиц обработки (n). Заполнение такого матрично-векторного блока без знаков двоичными цифрами может быть следующим.

При применении процедуры сжатия для такого блока положительным результатом уменьшения объема для хранения и передачи. Такой блок может быть в инверсном или прямом представлении. Следует отметить, что сама процедура сжатия и процедуры обратного развертывания должна быть простой и не занимать много времени. Алгоритм сжатия данных для матрично-векторного блока включает основные положения:

- матрично-векторный блок рассматривается как набор столбцов векторов - в каждом векторе ненулевые столбцы значимые единицы распределяются на три группы:

а) одиночные разряды;

б) группы по два разряда;

в) группы по три и более разрядов.

10101101010100101010101010101111100
10101010101111111001001010101010101
00101010101010111110100100101001010
10101010100000001000011111111111100
10001010010101010010010101010101000
.....
10100101010101010010101001010010101.

Для кодирования группы единичных разрядов необходимо бит, группы из двух разрядов бит, последней группы бит. Первичная общая длина такого блока определяется как . После введения процедуры сжатия длина будет определяться как сумма кодирования по каждому столбцу, дополнительных служебных разрядов, которые могут составлять вектор, в котором указаны нулевые или

столбец из единиц или другое. Для построения общего алгоритма кодирования рассмотрим несколько простых вариантов построения. Моделированием доказано, оптимальными является матрично-векторные блоки с параметрами m=8 и z равными 16,32,.... Если количество строк в матрице равно 8, то коды значащих единиц имеют значение от 000 до 111, то есть кодируются триадами. Коды значимых единиц записываются в единый последовательный файл, в котором каждые три разряда означают код ЗЕ, а различие по столбцам определяется по правилу - если значение кода ЗЕ записан код с меньшим значением ЗЕ, то последний принадлежит следующему столбцу.

Пусть нанесен блок данных, который переформатирован в матрицу, в которой количество строк *m*=8 и столбцов *n*=16. Значимые единицы распределены по столбцам с вероятностью 0,2.

000 0 1 0 0 0 0 1 0 0 1 0 0 1 0 1 0
001 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0
010 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0
011 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
101 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
110 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
111 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0.

Кодированный первый столбец имеет следующие коды 010, 100; второй столбец - 000, 110. Общий файл после сжатия состоит из следующих кодов
010 - 100 - 000 - 110 - 010 - 001 - 011 - 001 -110 - 010 - 000 - 100 - 001 - 110 - 010 - 111 - 000 - 011 - 010 - 001 - 111 - 000 - 001 - 000 - 010 – 011.

Длина такого файла 78 разрядов вместо 128 в исходном файле, т.е. уменьшение, составляет 40%. Рассмотрим матрицу (блок данных) в которой значимые единицы распределении с вероятностью до 0,4:

000 0 1 1 0 0 0 1 1 0 1 0 1 1 1 1 0
001 0 0 0 1 1 0 0 1 1 0 0 0 0 1 1 0
010 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0
011 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1
100 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
101 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0
110 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1
111 1 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0.

Первый столбец после кодирования представлено как 010, 100, 111, а второй 000, 011, 110. В восьмом столбце имеем 5 значимых единиц, поэтому лучше кодировать при инверсном представлении этого столбца -011 100 111. Файл после кодирования получим такой: 010 100 111 000 011 110 000 010 011 111 001 100 110 011 100 101 000 010 100 110 011 100 111 001 010 101 111 000 011 111 101 000 010 101 111 000 010 110 000 001 011 100 110 011 110.

Исходная матрица имеет в своем составе 128 элементов, а полученный файл 135 элементов (разрядов), т.е. имеем увеличение длины на 5%. Последний пример показывает, что эффективность сжатия по значимым единицам является зависимой от вероятности распределения ненулевых разрядов. В табл.1 приведены результаты моделирования для прямого и инверсного представления данных по столбцам, где fp-прямое представление блока, fi-инверсное представление блока.

Табл. 1. Результаты моделирования ля прямого и инверсного представления

Вероятность Заполнения блока	**00.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
fp, m =8, z=16	-48,6	-55	-51	-37	-21	2	30	58	87
fp, m =8, z=32	-51	-52	-48	-36	-19	1.8	29	56	87
fi, m =8, z=16	28	3	-20	-39	-51	-55	-48	-30	-36
fi, m =8, z=32	28	3,3	-19	-37	-48	-57	-55	-34	-38

Из результатов моделирования видно, что эффективным является одновременное кодирование данных по столбцам при инверсном и прямом представлении (особенно при вероятностях p =0,1-0,2-0,3). Общее правило кодировка не срабатывает, когда в столбцах матрицы значимые единицы распределены следующим образом

000 1 0
001 0 0
010 0 0
011 1 0
100 0 0
101 0 1
110 0 1
111 0 0.

При такой комбинации коды ЗЕ в соседних столбцах невозможно правильно распознать, поэтому надо вводить специальный служебный вектор с указаниями относительно номера столбца. Для варианта, который рассматривается (номер столбца - 000) (номер столбца - 001) 000 011 101 110.

Необходимо учесть и другие ситуации, которые нужно отразить в таком служебном векторе. - инверсное кодирование столбца;- исключительная ситуация; - „единичный" столбец; - „нулевой столбец.

Рассмотрим пример, когда в матрице встречаются строки, не содержащие единичных разрядов вообще. Это приводит к трудностям при декомпрессии, поскольку мы можем потерять целую группу байтов, если не введем дополнительную информацию в служебный вектор нулевой, либо единичный столбец. Для этого введем в служебный вектор дополнительное количество битов, что соответствует количеству строк нашей матрицы. При обнаружении нулевого (единичного) столбца переводим бит, отвечающий за данную строку в 1.

Кодированный первый столбец имеет следующие коды 010, 100; второй столбец - 000, 110. Общий файл после сжатия состоит из следующих кодов

010 - 100 - 000 - 110 - 010 - 001 - 011 - 001 -110 - 010 - 000 - 100 - 001 - 110 - 010 - 111 - 000 - 011 - 010 - 000 - 001 - 000 - 010.

000 0 1 0 0 0 0 1 0 0 1 0 0 1 0 1 0
001 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0
010 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0
011 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
100 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
101 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
110 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
111 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0.

При этом служебный вектор имеет вид 0 0000000000010001.

Первый бит служебного вектора служит для обозначения формы, в которой выполняется компрессия: 0 - прямое кодирование, 1 - инверсное кодирование. Остальные 16 бит соответствуют 16 столбикам нашей матрицы. Мы имеем нулевые столбцы с номерами 12 и 16, что отражено в нашем векторе.

Длина такого файла 86 разрядов, включая служебный вектор, вместо 128 в исходном файле, т.е. уменьшение, составляет 32%.

При выявлении единичных строк следует переходить к инверсного кодирования, и применять алгоритм, подобный обнаружение и кодирование нулевых столбцов.

Под исключительной комбинацией значимых единиц обозначается ситуация, когда не работает общее правило кодирования. Решить такую ситуацию возможно с помощью следующего правила.

Правило 1. Если в следующем столбце коды значимых единиц больше, чем в предыдущем следует добавить в предыдущий столбец код значимой единицы максимального значения, т.е. код последнего элемента.

При восьми строчной организации блока кодирования это 111. В служебном векторе при этом следует присвоить единичное значение соответствующем флага.

Таким образом, в служебном векторе должны быть *z* групп по два разряда. Первый разряд это флаг, определяющий прямое или инверсное представление в данном столбце, второй разряд - это флаг, определяющий наличие исключительной ситуации. Например, для блока данных будем иметь следующий служебный вектор и кодированный файл:

000 0 1 1 0 0 0 1 1 0 1 0 1 1 1 1 0
001 0 0 0 1 0 0 0 1 1 0 0 0 0 1 1 0
010 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0
011 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1
100 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
101 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0
110 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1
111 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0.

Служебный вектор

00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
010 000 000 001 100 010 000 000 001 000 000 000000
100 011 010 011 110
111 110 011 111
111

Результаты моделирования показывают, что объем данных после сжатия по предложенному алгоритму уменьшается в среднем до 10%. При показателях вероятности заполнения p=0,3-0,4 сжатие данных дает наименьший результат, но такой подход имеет внедрение на практике , потому что реальные показатели заполнения блоков данных являются различными.

Общая процедура при сжатии данных, при передаче и хранении в компьютерной среде может выполняться при следующих этапах:

1. Блоки данных преобразуются по предложенным алгоритмам сжатия.

2. По лучшим результатам сжатия осуществляется выбор соответствующего блока после применения алгоритмов.

3. В программу-монитор записывается необходимая информация.

4. В компьютерной среде, что принимает или сохраняет данные с помощью программы-монитора, выполняются процедуры по развертыванию данных в первоначальный вид.

Следующий метод сжатия данных при хранении и передачи данных основан на определении межразрядных расстояний – определении приращений при кодировании.

Структуры данных, которые хранятся в запоминающей среде компьютерных средств, а также предназначены для передачи между слотами компьютерной среды могут представлять собой линейные упорядоченные массивы. Например, последовательность следующего типа

A → 34 32 30 29 22 17 …..

и так далее. Таким массивом может быть и двоичная последовательность. Например, массив из 16 разрядов

A → 1101 0101 0101 0100

может быть преобразован в последовательность номеров ненулевых разрядов, где каждый номер изменяется в пределах размерности рассматриваемой последовательности. Массив *A* из 16 разрядов имеет вид

A* → 16 15 13 11 9 7 5 3.

Определим объём хранения таких данных. Выбираем максимальный номер из последовательности A* = 16 для хранения которого необходимо 5 разрядов – 10000. Поэтому последовательность A* занимает объём 40 бит:

10000 01111 01101 01011 01001 00111 00101 00011.

Для уменьшения объёма хранения-передачи используем принцип определения последовательности через межразрядные расстояния.

Для последовательности *A** такие расстояния P будут следующие:

P → 1 - 2 – 2 – 2 - 2 – 2 – 2.

Определим объём хранения таких P данных. Также выбираем максимальный размер – 2. Для его хранения необходимо два (2) разряда и, следовательно,

Общий размер хранения составляет 14 разрядов:

01 – 10 – 10 – 10 – 10 – 10 - 10.

Для точного восстановления последовательности *A** следует хранить один из первоначальных кодов – например младший 3 (11) и последовательность из 14 разрядов массива Р (01 – 10 – 10 – 10 – 10 – 10 - 10). Общий объём составляет 16 разрядов, что совпадает с исходным объемом *A* и меньше последовательности А* на (16/40) 60%.

При определенных соотношениях размерности исходного массива и количества ненулевых разрядов, в массиве рассматриваемые принципы хранения- передачи дают больший выигрыш по объёму – размерности преобразованного массива. Если же к исходному массиву применить предварительно код «-1», то выигрыш составит еще большую величину.

Например, массив из 32 разрядов

А → 0111 1100 1111 1110 1101 0111 1101 0100.

После применения кода «-1» имеет вид

1000 0101 0000 0010 1101 1000 0001 0100.

Массив номеров ненулевых разрядов

32 27 25 18 16 15 13 12 5 3.

Расстояния между номерами

5 – 2 - 7 - 2 – 1 – 2 – 2 - 7 - 2.

Для хранения такого массива требуется 27 бит - (101-010-111-010-001-010-010 – 111- 010) и для младшего номера 3 бита - 011. Общий объём - 30 бит, что меньше исходного объёма. Оптимальным является такое расположение разрядов, при котором межразрядные расстояния не являются большими числами, в противном случае выгодно передавать и хранить массив в виде номеров значащих разрядов.

Этапы метода хранения передачи с приращениями следующие:

1. Двоичный массив *A* представить в виде массива номеров значащих разрядов А*. При этом базовым параметром может быть размерность массива.

2. Вычислить межразрядные расстояния в *A** - массив *P*.

3. Определить максимальное значение в *P*.

4. Определить объём хранения- передачи *P* и сравнить полученное значение с исходным массивом *A*.

Моделирование метода приращений определяет уровни сжатия от 10% до 72%, что подтверждает высокую его эффективность.

Выводы

Методы сжатия данных на основе приращений и перекодирования матрично-векторных блоков имеют высокую эффективность, простые алгоритмы реализации, адаптированы к компьютерной среде. Недостатком является многовариантность выбора, что приводит к дополнительным временным затратам при реализации.

Список литературы

1. Гамаюн В.П. Разрядно-логарифмическая арифметика. Методы и алгоритмы. – К.: Книжное издательство НАУ, 2007. – 272 с.

2. Гамаюн В.П. Моделювання багаторозрядних комп'ютерних систем: Навч. посібник. – К.: Книжкове вид-во НАУ, 2007. – 112 с.

Статтю подано до редакції 30.09.2014