

Методы построения высокопроизводительных систем на основе сжатия данных¹

В.Н. Гридин, В.И. Анисимов, М.М. Абухазим

Аннотация. Приводится описание основных методов компактного хранения и обработки разреженных матриц и дается сравнительная оценка эффективности сжатия данных на основе списковых схем и методов фиксированного формата. Рассматриваются различные варианты списковых схем сжатия данных, и выполняется оценка их производительности. Отмечается, что практическая реализация методов Кнута связана с необходимостью полной перекодировки всех массивов при ведении в описание дополнительных элементов, что связано с большими техническими трудностями в процессе построения программного обеспечения и снижением производительности его дальнейшего функционирования. Показывается, что особенностью методов фиксированного формата является невозможность включения в описание дополнительных ненулевых элементов, что не позволяет применить эти методы для обработки информации при решении систем уравнений из-за неизбежного появления новых ненулевых элементов в процессе решения. Устанавливается общая структура двухэтапного формирования компактного описания и дается описание реализации сжатия данных на основе символического и численного анализа.

Ключевые слова: системы автоматизированного проектирования, веб-сервисы, моделирование систем, компактная обработка, разреженные матрицы, распределенные системы, Интернет-технологии.

Введение

Актуальной проблемой в области дальнейшего развития систем автоматизированного проектирования является реализация компактных методов обработки разреженных матриц, что позволяет существенно повысить эффективность программного обеспечения систем. Эта задача особенно актуальна при построении систем с распределенной архитектурой, когда информационные ресурсы предоставляются потребителям посредством сетевых сервисов [1-3]. При построении систем на основе Интернет-технологий существенно повышаются требования к быстродействию программного обеспечения, поскольку необходимо минимизировать время взаимодействия пользователя с системой.

Поэтому при создании систем автоматизированного проектирования необходимо использовать методы, основанные на формировании компактного описания моделируемой системы на основе выбранного способа сжатия данных, что повышает производительность обработки нулевых элементов в математическом описании системы. Компактное описание моделируемой системы при этом содержит сжатые массивы, ненулевые элементы которых определяются параметрами компонентов и в общем случае являются данными типа T , при этом тип T при моделировании во временной области определяется данными типа $double$, а при моделирование в частотной области – объектами некоторого класса $complex$.

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта 14-07-00112 а

Существующие методы компактной обработки разреженных матриц значительно отличаются своей эффективностью и поэтому для правильного выбора метода сжатия данных необходимо провести сравнительную оценку таких методов, которая приводится ниже.

В качестве критериев оценки приняты:

а) возможность введения в компактное описание дополнительных элементов, неизбежно появляющихся в процессе решения уравнений

б) коэффициент эффективности использования памяти $\beta = M/M_1$, где M – объем памяти, требуемый для полного описания разреженной матрицы, M_1 – объем памяти, требуемый для компактного описания разреженной матрицы.

1. Списковые схемы сжатия данных

К списковым схемам относится метод Кнута [4, 5], а также сокращенные варианты основного метода.

Для основного метода Кнута необходимо создать следующие компактные массивы:

WZ – для значений ненулевых элементов W_{ij} исходной матрицы, которые располагаются в этом массиве по относительному адресу a , имеющему значение порядкового номера элемента в массиве WZ;

WI – для номеров строк ненулевых элементов W_{ij} ;

WJ – для номеров столбцов ненулевых элементов W_{ij} ;

NR – для хранения относительного адреса a следующего ненулевого элемента W_{ij} строки;

NC – для хранения относительного адреса a следующего ненулевого элемента W_{ij} столбца;

ER – для хранения относительного адреса a начального элемента очередной строки;

ES – для хранения относительного адреса a начального элемента очередного столбца.

Достоинством метода Кнута является возможность записи элементов массива WZ в любом порядке, и как следствие, возможность введения в описание дополнительных ненулевых элементов, которые неизбежно появляются при решении уравнений. Следует также отметить, что классический метод Кнута позволяет

сканировать исходную матрицу, как по строкам, так и по столбцам, что является также достоинством метода. Вместе с тем, некоторым недостатком метода является значительное число массивов, необходимых для хранения информации о ненулевых элементах исходной матрицы.

Если учесть, что массивы WZ, WI, WJ, NR, NC имеют длину m , определяемую числом ненулевых элементов, а массивы ER, ES имеют длину n , определяемую порядком исходной матрицы, то эффективность использования памяти для данных T типа double, может быть определена для метода Кнута выражением:

$$\beta = \frac{8n^2}{16 * n^2 * \alpha + 4 * n} \cong \frac{1}{2 * \alpha}$$

Из полученного выражения видно, что с уменьшением α эффективность метода неограниченно возрастает.

Сокращенные методы Кнута позволяют уменьшить число используемых массивов путем исключения из полного описания ряда массивов. При этом возможно построение сокращенного метода, который реализует сканирование лишь по строкам, для чего необходимо ввести только массивы WZ, WJ, NR, ER, и построение сокращенного метода, который реализует сканирование только по столбцам, для чего необходимо использовать только массивы WZ, WI, NC, ES.

Эффективность любой из схем сокращенного метода Кнута может быть определена выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 12 + n * 2} \cong \frac{1}{1,5 * \alpha}$$

Очевидно, что эффективность сокращенной схемы Кнута выше, чем эффективность полной схемы Кнута, однако это достигается за счет ограничения возможности выбора направления сканирования, что является некоторым недостатком сокращенного метода Кнута. Однако возможность введения в описание дополнительных ненулевых элементов, а так же, произвольный порядок их записи в массиве остается в силе. Это обстоятельство является существенным достоинством сокращенного метода Кнута.

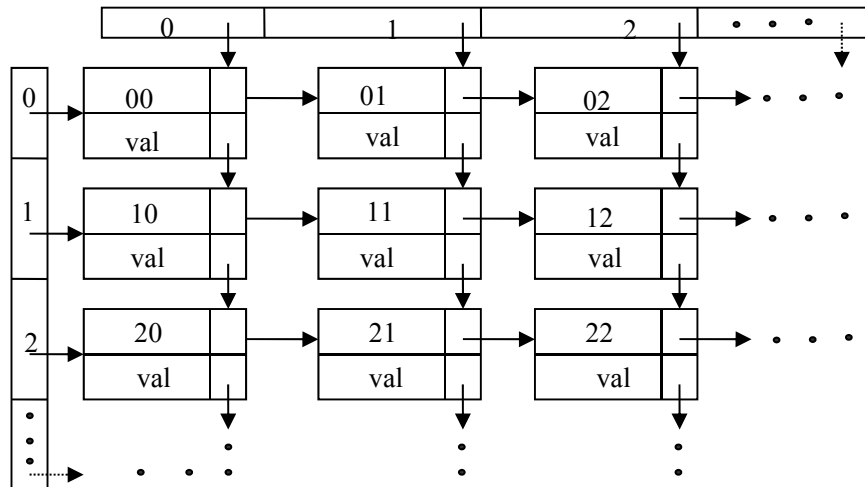


Рис. 1. Связь списков с объектами rows и columns

Вместе с тем практическая реализация методов Кнута связана с необходимостью полной перекодировки всех массивов при ведении в описание дополнительных элементов, что связано с большими техническими трудностями в процессе построения программного обеспечения и снижением производительности его дальнейшего функционирования. Возникающие проблемы могут быть устранены путем перехода к объектно-ориентированному подходу описания схемы Кнута на основе существующей в современных языках программирования C# и Java методики работы с коллекциями объектов класса ArrayList предназначенный для поддержки динамических массивов, размеры которых могут изменяться в процессе работы. Класс ArrayList реализует ряд интерфейсов и помимо свойств и методов, определенных в интерфейсах, имеет также и собственные свойства и методы [6, 7].

При этом для решения задачи построения компактного описания необходимо создание объектов rows и columns класса ArrayList. Каждый элемент объекта rows соответствует определенной строке матрицы и ссылается либо на ненулевой элемент в данной строке, либо указывает в null, если в строке элементов нет, а каждый элемент объекта columns соответствует определенному столбцу матрицы и ссылается либо на ненулевой элемент в данном столбце, либо указывает в null, если в столбце отсутствуют элементы [8, 9].

Описание элементов в списке задается в классе Element, члены которого val, i, j описывают значение элемента w_{ij} исходной матрицы и значения, определяющие его координаты i, j , а ссылки nextI, nextJ указывают на следующий элемент в строке и на следующий элемент в столбце соответственно.

```
class Element
{
    public double val;
    public int i;
    public int j;
    public Element nextI;
    public Element nextJ;
    public Element(double val, int i, int j)
    {
        this.val = val;
        this.i = i;
        this.j = j;
    }
}
```

Компактное описание всей разреженной матрицы выполняется в классе Matrix, структура которого имеет следующей вид:

```
public class Matrix
{ Element el;
  ArrayList rows = new ArrayList();
  ArrayList columns = new ArrayList();
  public Matrix(){}
  //Методы включения в список новых элементов
  //Методы для доступа к элементам списка
}
```

Реализация методов включения в список новых элементов и методов для доступа к элементам списка базируется на использовании стандартных методов, определенных в интерфейсах класса `ArrayList`.

Общая структура, отображающая связь списков с объектами `rows` и `columns`, приведена на Рис. 1.

Эффективность списковой схемы при использовании класса коллекций по сравнению с классической схемой Кнута несколько снижается, так как для каждого объекта класса `Element` помимо значений `val`, `i`, `j` необходимо также хранить значения ссылок `nextI` и `nextJ`. При этом эффективность полной схемы определяется выражением

$$\beta = \frac{8n^2}{24n^2\alpha + 8n} \cong \frac{1}{3\alpha}$$

Аналогичным образом могут быть реализованы и сокращенные схемы Кнута.

2. Сжатие данных в фиксированном формате

Сжатие данных на основе методов фиксированного формата возможно реализовать в строчном фиксированном формате и в строчно-столбцовом фиксированном формате [5].

Метод строчного фиксированного формата основан на использовании следующих массивов:

WZ – массив для хранения значения ненулевых элементов w_{ij} исходной матрицы.

WJ – массив для хранения индексов столбцов ненулевых элементов исходной матрицы W .

ER – массив, содержащий указатели точек входа в очередную строку.

Длина массивов WZ, WJ составляет m элементов, а длина массива ER составит $n + 1$ элементов, при этом в элемент $n + 1$ заносится значение $m + 1$.

Отличительной особенностью метода является невозможность произвольного выбора порядка записи ненулевых элементов в массиве WZ, и как следствие, невозможность включения в описание дополнительных ненулевых элементов. Это объясняется тем, что формат всех массивов жестко зафиксирован, и не мо-

жет меняться произвольным образом в процессе расчета.

Эффективность метода строчного фиксированного формата определяется выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 10(n + 1) * 2} \cong \frac{1}{1,25 * \alpha}$$

Очевидно, что эффективность метода выше, чем эффективность рассмотренных ранее методов. Существенным недостатком метода является невозможность включения в описание дополнительных элементов, что объясняется наличием фиксированного формата.

Метод строчно-столбцового формата также основан на использовании фиксированного формата, однако, в отличие от предыдущего метода, предполагается, что исходная матрица является структурно-симметричной, так что для каждого ненулевого элемента w_{ij} можно поставить в соответствие элемент w_{ji} . В случае если такой элемент в исходной матрице отсутствует, то его необходимо создать искусственно путем включения в компактное описание элемента $w_{ji} = 0$.

Для компактного хранения исходных элементов матрицы, требуется создать три массива:

WD – массив для хранения диагональных элементов,

WL – массив для хранения ненулевых элементов, расположенных ниже диагонали (поддиагональных элементов),

WU – массив для хранения ненулевых элементов, расположенных выше диагонали (наддиагональных элементов).

Массив WD должен содержать n диагональных элементов, в случае нулевого значения диагонального элемента в исходной матрице его значение $w_{ii} = 0$ должно учитываться в массиве WD.

В соответствие с методом при формировании массива WU наддиагональные элементы записываются по строкам, а при формировании массива WL поддиагональные элементы записываются по столбцам. Согласно принятому порядку формирования массивов WU, WL, относительный адрес некоторого элемента w_{ij} ,

расположенного в массиве WU, совпадает с относительным адресом элемента w_{ji} , расположенном в массиве WL. Отмеченное свойство существенно упрощает процесс программирования для организации сканирования элемента.

Для хранения индексов строк и столбцов ненулевых элементов в методе строчно-столбцового фиксированного формата используется массив WJI, при этом этот массив содержит номера столбцов ненулевых элементов, расположенных выше диагонали, которые совпадают с номерами строк транспонированных ненулевых элементов, расположенных ниже диагонали. Для определения точки входа в строку выше диагонали (точка входа в столбце ниже диагонали) используется массив ERC. В последний n -ый элемент этого массива заносится значение $\frac{m-n}{2} + 1$.

Если учесть, что массив WD имеет n элементов, длина массивов WU, WL, WJI составляет $\frac{m-n}{2}$ элементов, а длина массива ERC составляет n элементов, то эффективность метода строчно-столбцового фиксированного формата может быть определена выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 9 + n} \cong \frac{8}{9} = \frac{1}{1,1 * \alpha}.$$

Отсюда следует, что эффективность метода выше, чем всех рассмотренных ранее методов. Однако, так же как для метода строчного фиксированного формата, метод строчно-столбцового фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке, следовательно, не имеется возможность включения дополнительных элементов в компактное описание.

Поскольку формат всех массивов жестко зафиксирован, и не может меняться произвольным образом в процессе расчета, отличительной особенностью метода является невозможность включения в описание дополнительных ненулевых элементов. Это обстоятельство исключает возможность непосредственного применения строчно-столбцового фиксированного формата для обработки информации при решении систем уравнений любым численным ме-

тодом вследствие неизбежного появления новых ненулевых элементов в процессе этого решения.

Поэтому для возможности полноценного использования такого метода компактной обработки данных на всех шагах процесса моделирования необходима его модификация путем введения двухэтапной процедуры формирования компактного описания, включающей в себя символьный этап и численный этап [10]. Целью символьного этапа является только определение размеров всех используемых массивов с учетом появления новых ненулевых элементов.

Так как на символьном этапе не ставится задача формирования численных массивов, то его выполнение осуществляется путем обработки некоторой индексной матрицы С, элементы которой имеют только два произвольных значения (например, 0 и 1). При этом, если в исходной матрице некоторый элемент $w_{ij} \neq 0$, то для введенной индексной матрицы С соответствующий элемент $C_{ij} = 1$, а все остальные элементы равны 0. Следовательно, вместо рассмотрения исходной схемы, может рассматриваться некоторый «портрет» этой схемы, отображающий ее структуру, но не содержащий информации о численных значениях параметров.

Формирование численных массивов осуществляется на втором этапе, при выполнении которого используются определенные на первом этапе форматы массивов, учитывающие зарезервированные места для всех новых ненулевых элементов. Для формирования компактных численных массивов, а также для решения систем уравнений на основе LU-факторизации, разработаны виртуальные алгоритмы формирования и обработки компактных массивов без построения полного математического описания задачи.

Существенным достоинством такой двухэтапной процедуры является разделение ее на две независимые части символьного и численного анализа. Так как практически все реальные задачи проектирования электронных схем связаны с многовариантным расчетом схемы одной и той же структуры, то символьный этап выполняется для каждой структуры единственный раз, в то время как численный этап реализуется десятки, сотни, а часто и тысячи раз.

Поэтому накладные расходы от введения символического этапа в реальных задачах моделирования систем с разреженными матрицами оказываются весьма незначительными, так как все объекты, связанные с обработкой индексной матрицы S , не используются при выполнении численного этапа. Поскольку программное обеспечение современных распределенных систем автоматизированного проектирования реализуется на платформенно-независимых языках Java или C#, имеющих встроенные средства распределения и освобождения неиспользуемой динамической памяти, то все объекты, созданные на символическом этапе, автоматически удаляются системой при завершении этого этапа.

Заключение

При создании систем автоматизированного проектирования необходимо использовать методы формирования компактного описания моделируемой системы на основе выбранного способа сжатия данных, что повышает производительность обработки нулевых элементов в математическом описании системы. Из возможных методов сжатия данных для решения этой задачи следует рекомендовать метод строчно-столбцового фиксированного формата, поскольку этот метод выгодно отличается от других методов своей эффективностью. Для возможности коррекции формата при появлении новых ненулевых элементов следует использовать двухэтапную процедуру, основанную на предварительном построении символического портрета моделируемой схемы с последующим численным расчетом схемы в реальном формате.

Гридин Владимир Николаевич. Директор Центра информационных технологий в проектировании РАН. Окончил Московский авиационный технологический институт в 1972 году. Доктор технических наук, профессор. Автор 200 печатных работ. Область научных интересов: информационные технологии в проектировании, безопасность систем.
E-mail: info@ditc.ras.ru

Анисимов Владимир Иванович. Профессор Санкт-Петербургского электротехнического университета. Окончил Ленинградский электротехнический институт в 1949 году. Доктор технических наук. Автор 250 печатных работ. Область научных интересов: моделирование систем, распределенные системы автоматизированного проектирования.
E-mail: vianisimov@inbox.ru

Абухазим Монзер Мохаммед. Аспирант Санкт-Петербургского государственного электротехнического университета. Окончил Санкт-Петербургский государственный электротехнический университет в 2013 году. Область научных интересов: моделирование систем, веб-ориентированные системы автоматизированного проектирования.
E-mail: Abuhazim_monther@Yahoo.com.

Практическая реализация двухэтапного процесса сжатия данных позволяет существенно повысить эффективность работы распределенных систем автоматизированного проектирования.

Литература

1. Гридин В.Н., Анисимов В.И. Методы построения систем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц. // Информационные технологии в проектировании и производстве. №1, 2009.
2. Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение систем автоматизированного проектирования на основе Web-сервисов. // Автоматизация в промышленности, №1, 2011.
3. Коваленко О.С., Курейчик В.М. Обзор проблем и состояний облачных вычислений и серверов. // Известия ЮФУ. Технические науки. № 7, 2012.
4. Кнут Д. Искусство программирования для ЭВМ, т. 1, пер. с англ. - М.: Мир, 1976, 734 с.
5. Писсанецки С. Технология разреженных матриц, пер. с англ. - М.: Мир, 1988, 406 с.
6. Троелсон Э. Язык программирования C# 2005 и платформа .NET 2.0, пер. с англ. - М.: Изд. «Вильямс» 2007, 1167 с.
7. Ноутон П., Шилдт Г. Java 2, пер. с англ. - СПб.: Изд. «БХВ-Петербург» 2001, 1072 с.
8. Анисимов В.И., Амахвр Ю.М. Компактные методы обработки разреженных матриц задач мониторинга на основе списочных структур. // Труды Пятой Международной конференции «Приборостроение в экологии и безопасности человека» - СПб.: ГУАП, 2007, 316 с.
9. Амахвр Ю.М. Компактная обработка разреженных матриц на основе списков. // Вестник компьютерных и информационных технологий. №4, 2008.
10. Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение веб-сервисов систем автоматизации схемотехнического проектирования. // Информационные технологии и вычислительные системы. №4, 2012.