

K. A. Borisenko, Ya. A. Bekeneva, N. N. Shipilov, A. V. Shorov
Saint-Petersburg state electrotechnical university «LETI»

HYBRID MODELING SYSTEM FOR DDoS ATTACKS PROTECTION METHODS DEVELOPMENT AND TESTING

Protection against distributed attacks «distributed denial of service» (DDoS) is a very important task nowadays. In the paper we introduce a hybrid system for simulating DDoS attacks and computer network protection techniques. The developed system makes it possible to create various network topologies, perform experiments with DDoS attack simulation, develop new protection methods and test the existing ones. The suggested system not only allows us to design virtual networks, but also makes it possible to connect real network nodes for improving the accuracy of the experiments. The paper contains information on component formal models and system architecture. We also describe virtual network verification relative to a real network. Furthermore, we provide the experiments devoted to DDoS attacks and operation of the protection techniques.

DDoS attack, virtual network, real network, simulation modeling, client behavior scenario, protection techniques, OMNeT++, INET, ReaSE, PlanetLab, SYN Flooding, HTTP Flooding, Egress Filtering

УДК 519.87

В. И. Анисимов, М. А. Шабани
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Повышение эффективности систем автоматизированного проектирования на основе компактных методов обработки разреженных матриц

Рассматриваются методы повышения эффективности систем автоматизированного проектирования на основе технологии разреженных матриц. Приводится описание реализации компактных методов обработки разреженных матриц при построении программного обеспечения системы автоматизированного проектирования

Системы автоматизированного проектирования, моделирование систем, разреженные матрицы, распределенные системы, интернет-технологии

Одной из актуальных проблем в области дальнейшего развития систем автоматизированного проектирования является внедрение в них компактных методов обработки разреженных матриц, что позволяет существенно повысить эффективность программного обеспечения систем. Такая задача становится особенно актуальной при построении систем с распределенной архитектурой, где информационные ресурсы предоставляются потребителям посредством сетевых сервисов [1]–[3]. Использование интернет-технологий при разработке систем автоматизированного проектирования дает возможность реализовать слабосвязанное программное обеспечение, вследствие чего взаимодействие между приложе-

ниями не нарушается каждый раз, когда меняется реализация какого-либо сервиса, а также обеспечить взаимодействие на любой платформе между различными приложениями, написанными на любом языке программирования. Вместе с тем при построении систем с распределенной архитектурой существенно повышаются требования к быстродействию программного обеспечения, поскольку требуется минимизировать время взаимодействия пользователя с системой.

В связи с этим при создании систем автоматизированного проектирования необходимо использовать методы, основанные на формировании компактного описания моделируемой системы на основе тех или иных способов сжатия данных,

повышающих производительность обработки нулевых элементов в математическом описании моделируемой системы. Компактное описание моделируемой системы при этом содержит сжатые массивы, ненулевые элементы которых определяются параметрами компонентов и являются в зависимости от характера решаемой задачи данными типа `double` (моделирование во временной области) или `complex` (моделирование в частотной области).

Возможные методы компактной обработки разреженных матриц существенно отличаются по своей эффективности, и поэтому для правильного выбора метода сжатия данных необходимо провести их сравнительную оценку. 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, требующийся для полного описания разреженной матрицы; M_1 – объем памяти, необходимый для компактного описания разреженной матрицы.

При этом предполагается, что выполняется моделирование в частотной области и используется тип данных `complex`, реализованный на основе пользовательского или встроенного класса, в котором как для вещественной, так и для мнимой части комплексного числа используется тип данных `double` и, следовательно, объем памяти $M = 16n^2$.

Связные схемы хранения. К связным схемам относится классический метод Кнута [4], а также сокращенные варианты основного метода.

При использовании основного метода Кнута необходимо создать следующие массивы:

WZ – для значений ненулевых элементов w_{ij} исходной матрицы;

WI – для номеров строк ненулевых элементов;

WJ – для номеров столбцов ненулевых элементов;

NR – для хранения относительного адреса a следующего ненулевого элемента строки (a – порядковый номер элемента в массиве WZ);

NC – для хранения относительного адреса a следующего ненулевого элемента столбца;

ER – для относительного адреса a входа в очередную строку;

ES – для относительного адреса a входа в очередной столбец.

Весьма важная особенность метода Кнута – возможность записи элементов массива WZ в любом порядке и, как следствие, возможность введения в описание дополнительных ненулевых элементов. Достоинством метода Кнута является возможность сканирования исходной матрицы как по строкам, так и по столбцам, а недостатком – 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 для метода Кнута эффективность использования памяти можно определить выражением

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

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

Сокращенный метод Кнута позволяет уменьшить число используемых массивов исключением из полного описания массивов, которые реализуют сканирование по столбцам. При этом для компактного описания исходной матрицы необходимо ввести только массивы WZ, WJ, NR, ER. Аналогично можно составить вариант сокращенной схемы Кнута, позволяющий осуществить сканирование только по столбцам. При этом для компактного описания необходимо использовать только массивы WZ, ES, NC, WI.

Эффективность любой из схем сокращенного метода Кнута можно определить выражением

$$\beta = \frac{8n^2}{12n^2\alpha + 2n} \cong \frac{1}{1.5\alpha}.$$

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

Методы фиксированного формата. К методам фиксированного формата относятся метод строчного формата и метод строчно-столбцового формата.

Для использования метода строчного фиксированного формата [4] требуются следующие массивы:

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

WJ – для хранения индексов столбцов ненулевых элементов исходной матрицы W ;

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

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

Одной из отличительных особенностей 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 может меняться произвольным образом в процессе расчета.

Эффективность метода строчного фиксированного формата можно определить выражением

$$\beta = \frac{8n^2}{10n^2\alpha + (n+1)2} \cong \frac{1}{1.25\alpha}.$$

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 [5]–[7] также основан на использовании фиксированного формата, однако, в отличие от предыдущего метода, предполагается, что исходная матрица является структурно-симметричной, так что каждому ненулевому элементу w_{ij} можно поставить в соответствие элемент w_{ji} . Если такой элемент в исходной матрице отсутствует, то его необходимо создать искусственно включив в компактное описание элемент $w_{ji} = 0$.

Для компактного хранения исходных элементов матрицы требуется создать 3 массива:

WD – для хранения диагональных элементов;

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

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

В соответствии 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} , расположенного 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, содержащий 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 -й элемент этого массива заносится значение $\frac{m-n}{2} + 1$.

Если учесть, что длина массива WD составляет 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 можно определить выражением

$$\beta = \frac{8n^2}{9n^2\alpha + n} \cong \frac{1}{1.1\alpha}.$$

Отсюда следует, что эффективность метода выше, чем всех рассмотренных ранее методов. Однако, как и для метода строчного фиксированного формата, метод строчно-столбцового фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке, а следовательно, невозможно включить дополнительные элементы в компактное описание.

Поскольку формат всех массивов жестко зафиксирован и не может меняться произвольным образом в процессе расчета, отличительной особенностью метода является невозможность вклю-

но-адресной матрицы A используется тип данных длиной 2 байт, а для хранения каждого значащего элемента исходной матрицы используются данные длиной 8 байт, можно определить выражением

$$\beta = \frac{8n^2}{n^2(8\alpha + 2)} = \frac{1}{\alpha + 0.25}.$$

Из приведенного соотношения видно, что коэффициент экономии памяти для данного метода увеличивается с уменьшением коэффициента разреженности α .

При практической реализации задачи построения программного обеспечения для работы с индексно-адресными матрицами целесообразно применить для ее решения существующую в современных языках программирования C# и Java технологию работы с коллекциями [8], [9] и использовать класс ArrayList, предназначенный для поддержки динамических массивов, размеры которых могут изменяться в процессе работы. Класс ArrayList реализует интерфейсы ICollection, IList, IEnumerable, ICloneable и помимо свойств и методов, определенных в интерфейсах, имеет также и собственные свойства и методы. Для этого класса необходимо создать 2 объекта – для параметров схемы (например, WZ) и для задающих источников (например, SZ), ссылки на которые удобно объявить в специально созданном классе глобальных переменных (например, GV). В классе глобальных переменных следует также объявить ссылку типа short на индексно-адресный массив A. Сами объекты создаются в произвольной функции после ввода значений всех переменных для размерности схемы.

При построении программного обеспечения для работы с индексно-адресной матрицей необходимо решать две основные задачи. Первая задача связана с формированием компактного описания системы, вторая – с решением уравнений на основе созданного компактного описания.

Целью формирования компактного описания системы является занесение в массивы WZ и SZ некоторых значений, определяемых параметрами компонентов и структурой схемных связей. Так, если в компактный массив WZ необходимо занести некоторое значение z , которое в полном описании разреженной комплексной матрицы должно было бы заноситься в i -ю строку и j -й столбец этой матрицы, то при занесении в

компактный массив WZ такая задача решается вызовом функции $\text{set}(i, j, z)$, которая имеет описание

```
void set(int i, int j, Complex z)
{ int a = GV.A[i, j];
  if (a != 0)      GV.WZ[a] =
(Complex)GV.WZ[a] + z;
  else
  { GV.WZ.Add(z);
    GV.A[i, j] = (short)(GV.WZ.Count - 1); }
}
```

Аналогичным образом формируется и компактный массив SZ.

При решении уравнений любым из известных методов (метод Гаусса, Гаусса–Жордано, LU-факторизация) при использовании полного описания разреженной матрицы w необходимо выполнять k -й шаг приведения к треугольной матрице

$$w[i, j] = w[i, j] - w[i, k] * w[k, j] / w[k, k].$$

При использовании компактного описания такая задача решается вызовом функции $\text{step}(i, j)$, которая имеет описание

```
void step(int i, int j)
{ if (GV.A[i, j] == 0)
  { GV.WZ.Add(new Complex(0, 0));
    GV.A[i, j] = (short)(GV.WZ.Count - 1); }
  GV.WZ[GV.A[i, j]] = (Complex)GV.WZ[GV.A[i, j]] - (Complex)GV.WZ[GV.A[i, k]] *
(Complex)GV.WZ[GV.A[k, j]] / (Complex)GV.WZ[GV.A[k, k]];
}
```

В результате выполнения этой функции в массиве WZ будет выполнен k -й шаг виртуального преобразования к треугольной матрице. Аналогичным образом решается задача преобразования массива SZ.

Рассмотрение методов сжатия данных на основе компактной обработки разреженных матриц позволяет сделать вывод, что наибольший интерес для практической реализации высокопроизводительного программного обеспечения распределенных систем автоматизации схмотехнического проектирования представляет метод индексно-адресных матриц. Этот метод позволяет наиболее просто привести программное обеспечение к требуемой компактной форме, обеспечивающей возможность учета новых ненулевых элементов в компактном описании моделируемой системы.

СПИСОК ЛИТЕРАТУРЫ

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. 2009. № 1. С. 3–7.
2. Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А. Построение систем автоматизированного проектирования на основе Web-сервисов // Автоматизация в промышленности. 2011. № 1. С. 9–12.
3. Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А. Построение систем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. № 5. С. 23–27.
4. Писсанецки С. Технология разреженных матриц / пер. с англ. М.: Мир, 1988. 406 с.
5. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем / пер. с англ. М.: Радио и связь, 1988. 560 с.
6. Гридин В. Н., Анисимов В. И., Шабани М. А. Методы построения высокопроизводительных распределенных систем автоматизации схемотехнического проектирования // Информационные технологии. 2014. № 8. С. 59–63.
7. Гридин В. Н., Анисимов В. И., Шабани М. А. Моделирование систем на основе технологии разреженных матриц // Системы высокой доступности. 2014. № 4. С. 88–93.
8. Троелсон Э. Язык программирования C# 2005 и платформа .NET 2.0 / пер. с англ. М.: Вильямс, 2007. 1167 с.
9. Ноутон П., Шилдт Г. Java 2 / пер. с англ. СПб.: БХВ-Петербург, 2001. 1072 с.

V. I. Anisimov, M. A. Shabani

Saint-Petersburg state electrotechnical university «LETI»

IMPROVING THE EFFICIENCY OF CAD SYSTEMS BASED ON COMPACT METHODS OF SPARSE MATRICES

The methods of increasing the effectiveness of computer-aided design technology based on sparse matrices. Describes the implementation of the compact processing methods for sparse matrices in the construction of software computer aided design.

Systems of the automated designing, modeling of the systems, sparse matrices, the distributed systems, internet-technologies

УДК 159.9 + 303.732

Е. В. Андреевский, Е. А. Бурков, Н. А. Назаренко, П. И. Падерно
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Анализ стратегий профессионального психологического отбора (модели и характеристики)

Выделен ряд типовых структур стратегий профотбора, рассмотрен подход к их формальному описанию на основе многомерной модификации обобщенного структурного метода. Предложенные модели основных стратегий профотбора позволяют оценивать различные аспекты безошибочности профессионального отбора и выявлять оптимальную стратегию для каждой конкретной цели.

Профотбор, типовая стратегия, структура, обобщенный структурный метод, рабочая операция, контрольная операция, характеристики

Одним из ключевых элементов обеспечения высокой эффективности эксплуатации сложных человекомашинных комплексов (ЧМК) различного назначения является высокий профессионализм специалистов, работающих с ними, что можно обеспечить только на основе профессио-