Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования
Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчет по лабораторной работе

«Умножение разреженных матриц. Элементы комплексного типа. Формат хранения матрицы – столбцовый (CCS)»

Выполнил:
студент группы 381708-1
Шеметов Ф.А.

Проверил:
доцент кафедры МОСТ,
кандидат технических наук
Сысоев А. В.

Нижний Новгород
2020

Оглавление

Введение

Разрежённой матрицей называются матрицы, которые преимущественно состоят из нулевых элементов. Возникают разрежённые матрицы при решение различных задач связанных с научной и инженерной областью, где присутствует большое число неизвестных, связанных между собой уравнениями.

Так как в разрежённых матрицах содержится большинство значений в виде нулей, то любая арифметическая операция с этими матрицами увеличивает лишние вычислительные затраты. Для того чтобы снизить вычислительные затраты, были придуманы разные способы хранений разрежённых матриц и один из них — Формат хранения CCS[1].

Формат хранения CCS представляет собой структуру данных, которая позволяет хранить матрицу в виде трех массивов. Первый массив хранит ненулевые значения элементов матрицы. Второй массив хранит номера строк для каждого ненулевого элемента. Третий массив хранит индекс начала каждого столбца.

Сам формат хранения CCS предоставляет минимальные требования к памяти и значительно уменьшает вычислительные затраты при арифметических операциях с разрежёнными матрицами.

Цель лабораторной работы — изучение принципа хранения и разработка алгоритма умножения разрежённых матриц в формате CCS с использованием технологий параллельного программирования.

[1] Compressed Column Storage

Постановка задачи

В данной лабораторной работе ставится задача в виде разработки нескольких проектов, где нужно реализовать алгоритм умножения разрежённых матриц в формате хранения CCS с элементами комплексного типа.

Проект будет включать в себя:

- Набор юнит-тестов использующие Google C++ Testing Framework.
- Исходный и заголовочный файл. В заголовочном файле реализован интерфейс класса, а в самом исходном файле реализован код последовательного алгоритма умножения разрежённых матриц в формате хранения CCS или параллельный алгоритм с технологиями OpenMP, TBB, std::threads.
- файл CMake для сборки проекта.

Описание алгоритма

В нашей лабораторной работе нужно реализовать алгоритм умножения разрежённых матриц в формате хранения CCS с элементами комплексного типа.

Чтобы реализовать алгоритм умножения, нужно сгенерировать саму разрежённую матрицу. Для этого мы будем использовать ГПСЧ *mt19937*, который позволяет генерировать оптимально-хорошую последовательность случайных чисел. Так как мы реализовываем не плотную, а разрежённую матрицу которая преимущественно состоит из нулей, то для ГПСЧ нужно задать параметр в виде вещественного числа от 0 до 1...

После полученного сгенерированного класса разрежённой матрицы, нам нужно реализовать хранение этой матрицы в формате CCS. Но так как перенос из разрежённой матрицы в формат хранения CCS увеличивает вычислительные затраты, то для их уменьшения, мы сначала должны транспонировать матрицу и перевести в формат хранения CRS, получая формат хранения CCS, но возникает проблема в том, что разрежённая матрица может быть больших размеров, где большинство значений это нули, поэтому транспонирование этой матрицы будет тоже для нас затратным делом. И чтобы снизить затраты, нужен алгоритм транспонирования, который сразу преобразует матрицу из формата хранения CRS в формат хранения CCS.

Поэтому, первым делом мы преобразуем разрежённую матрицу в CRS формат. Все ненулевые значения, обходя последовательно по строкам матрицы, передаём в первый вектор(`value`), во второй вектор(`col_index`) мы передаём номера столбцов ненулевых значений матрицы, в третий вектор(`row_offset`) передаём индексы начала новых строк во втором векторе. Получив объект класса разрежённой матрицы в формате CRS, следующий шаг это алгоритм транспонирования CRS в CCS. Сформируем новый объект класса `obj` с пустыми векторами, в первом векторе(`obj.value`) также будут ненулевые значение, во втором(`obj.row_index`) уже будут номера строк, а в третьем(`obj.col_offset`) индексы начала новых столбцов. Сначала мы заполняем третий вектор нулями и инкрементируем значение этого вектора, где индекс является номер столбца. Получив заполненный третий вектор, мы обращаемся к нему и получаем индекс к вектору `obj.col_offset` и `obj.value`, где по-этому индексу во втором векторе(`obj.row_index`) в элементы передаём номера строк, а в первом(`obj.value`) в элементы передаём значения элементов первого вектора формата хранения CRS.

Получив объект класса разрежённой матрицы в формате хранения CCS, мы теперь можем реализовать алгоритм умножения CCS матриц. Умножение двух CCS матриц происходит по алгоритму JKI-типа.

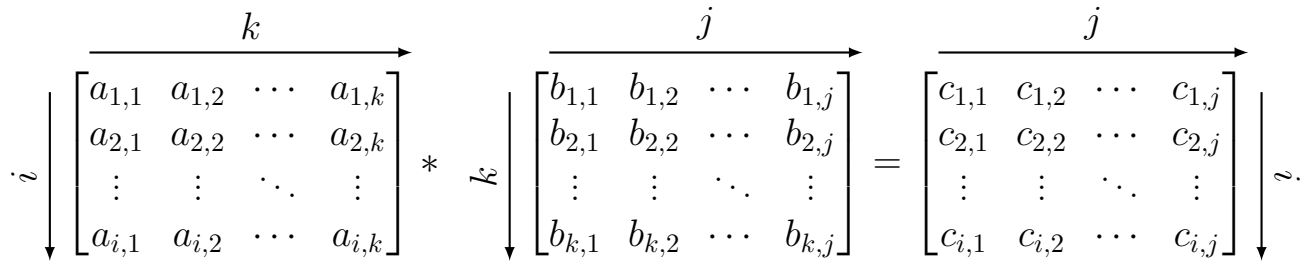

Рис. 0.1. Умножение разрежённых матриц в формате хранения CCS. Постолбцовый JKI алгоритм.

Возьмём матрицу A и B в формате хранения CCS. Начинаем обходить столбцы матрицы B. Взяв столбец, идём по вектору `col_offset` и получаем индекс расположения ненулевых элементов в строках. После чего поэтому индексу определяем номер столбца матрицы A и циклом обходим строки этого столбца, содержащие ненулевые элементы. Чтоб записывать результаты умножения, создаём новый вектор `tempDataVec`, где его индекс это номер строки столбца вектора A с ненулевыми элементами. Записываем результат умножения и суммирования для каждого элемента столбца матрицы B на весь столбец матрицы A в каждый индекс вектора `tempDataVec`, где заполненный вектор является столбцом исходной матрицы C. После этого, обходим циклом этот вектор, проверяем на отсутствие нулевых значений и передаём его элементы в первый вектор матрицы C, во второй число ненулевых значений в векторе `tempDataVec`, а в третий вектор размер первого вектора. В итоге получаем умноженную исходную матрицу C в формате CCS.