Владислав Страшко

2 марта 2020 г.

1 Постановка задачи

Необходимо вычислить на GPU сумму следующего вида:

$$C \sum_{i<j} f_{ij} \frac{q_i q_j}{r}$$

Здесь q_i — заряд соответствующего атома, r_i — расстояние между атомами,

$$f_{ij} = \begin{cases} 1, & \text{если атомы связаны более чем 3 связями или не связаны} \\ 1/2, & \text{если атомы связаны 3 связями} \\ 0 & \text{если атомы связаны менее, чем 3 связями} \end{cases}$$

2 Наивная реализация

Заметим, что если у нас есть матрица F, по диагонали которой стоят нули, а $F_{ij} = f_{ij}/r_{ij}$ при $i \neq j$, то ответ будет равен $\frac{C}{2} q^T F q$.

Таким образом, перед нами стоит задача нахождения путей длины не более 3 графе, вершины которого — атомы, а ребра — наличие связи между двумя атомами. Саму матрицу F можно вычислить как $\mathbb{I} - 0.5\,\mathrm{sign}(A^3 + A^2 + A) - 0.5\,\mathrm{sign}(A^2 + A)$, где $\mathbb{I}$ — матрица, состоящая полностью из единиц, A — матрица смежности графа. Это решение следует из факта, что если возвести матрицу смежности в n степень, то получившаяся матрица будет содержать на (i, j) позиции количество путей длины ровно n из i в j. Тогда, например, $A^3 + A^2 + A$ содержит количество путей длины более 3.

Наивная реализация состоит в том, чтобы явным образом возводить матрицу A во вторую, а затем и в третью степень, стандартным образом умножая матрицы. Но, у этого подхода, очевидно. Но, это очень трудоемкий как по памяти, так и по вычислительным ресурсам алгоритм. Матрицы умножаются за $O(N^3)$, при этом занимают $O(N^2)$ памяти.

3 Оптимизации

Обратим внимание на вид матрицы смежности. Она очень разреженная, и умножать ее как Dense-матрицу крайне неразумно. Для того, чтобы гораздо эффективнее производить операции над Sparse матрицами, есть разные подходы. В данном случае можно воспользоваться вариантом ELLPACK, описанном в Статье. Этот подход отлично подходит для хранения и операций с матрицами, где в каждой строке примерно одинаковое количество элементов. В нашем случае их максимум 4. Тогда эту матрицу можно хранить как $N \times K$, где K — максимальное количество элементов в строке. Кроме того, при реализации умножения на вектор, получается coalesced доступ к памяти. При этом в нашем случае стоимость умножения на вектор снижается до $O(N)$.

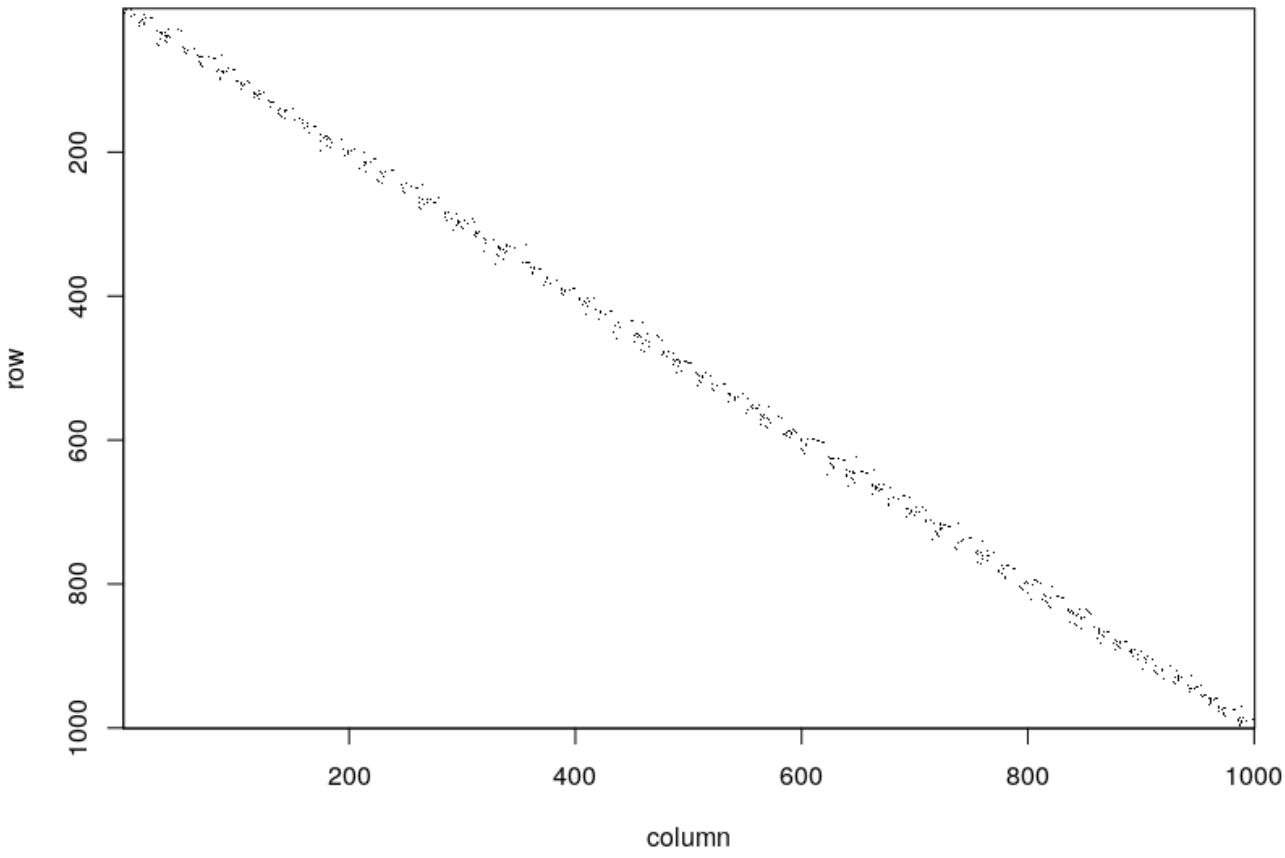

Рис. 1: Матрица смежности графа молекулы

4 Итоги

На данный момент реализован простейший вариант, с умножением матриц за куб (при этом без разделяемой памяти, поэтому доступ к памяти не coalesced). Он плох в данной задаче. Кроме того можно обобщить умножение sparse матрицы на вектор на случай матрицы-на-матрицу, как N умножений на вектор.