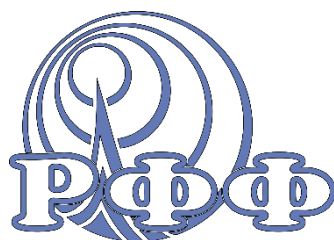


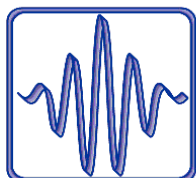


Национальный
исследовательский

**Томский
государственный
университет**



Радиофизический факультет



9-я Международная научно-практическая конференция
**Актуальные проблемы радиофизики
АПР-2021**

Сборник трудов конференции

при поддержке:



20-22 октября 2021 года
г. Томск

Тензорный подход к реализации случайных блочных клеточных автоматов: реализация в рамках TensorFlow

Шалыпина Наталья Андреевна

Громов Максим Леонидович

Торгаев Станислав Николаевич

Матолыгин Арсений Константинович

Национальный исследовательский Томский государственный университет

E-mail: nat.shalyapina@gmail.com

Клеточный автомат (КА) представляет собой дискретную динамическую модель, функционирование которой происходит в n -мерном пространстве, представляющим собой множество ячеек одинаковой формы и размера, и дискретные промежутки времени, называемые итерациями. Каждая ячейка модельного пространства соответствует некоторой области моделируемого физического пространства и описывается набором безразмерных характеристик. Основным преимуществом КА и математических моделей, построенных на их основе, является параллелизм, заложенный в структуру автоматов. Такая особенность позволяет привлекать современные суперкомпьютеры, состоящие из большого количества ядер, для построения эффективных программных реализаций.

Формально двумерный синхронный клеточный автомат представляет собой множество $\{A_{ij} | (i,j) \in \mathbb{Z} \times \mathbb{Z}\}$, где $A = \{S, \hat{s}, I, O, \varphi, \psi\}$ – множество автоматов Мура [1]: S – конечное непустое множество состояний автомата с выделенным начальным состоянием $\hat{s} \in S$, I – конечное множество входных воздействий, O – конечное множество выходных реакций, $\varphi: S \times I \rightarrow S$ – полностью определённая функция переходов, $\psi: S \rightarrow O$ – полностью определённая функция выходов. Каждый отдельный автомат Мура A_{ij} называется клеткой. Каждой паре (i, j) поставлено в соответствии некоторое конечное множество пар целых чисел $N_{ij} \subseteq \mathbb{Z} \times \mathbb{Z}$, называемое окрестностью пары (i, j) . Пары из N_{ij} называются соседями пары (i, j) .

Правила перехода клеток из одного состояния в другое (функция φ), тип соседства клеток (правила формирования множеств N_{ij}), число различных возможных состояний клеток (множество S) задают все многообразие синхронных двумерных клеточных автоматов [2].

Отдельного внимания заслуживает так называемый *блочный клеточный автомат с окрестностью Марголуса* [3] (рисунок 1). Этот автомат используется для построения модели диффузии. В модели поле автомата разбивается на два подмножества - блока: $Even = \{A_{ij}, A_{ij+1}, A_{i+j}, A_{i+1j+1}\}$, $Odd = \{A_{i+1j+1}, A_{i+1j+2}, A_{i+2j+1}, A_{i+2j+2}\}$. Правила перехода таковы, что на каждом такте работы автомата определяется направление поворота блока по или против часовой стрелки равновероятно ($p_1 = p_2$), либо с вероятностью $p_3 = (1 - p_1 + p_2)$ поворота не происходит. Автомат функционирует в синхронном двухтактном режиме: на чётных тактах правила перехода примеряются к чётным блокам, на нечётных тактах – к нечётным. Изменяя p_1, p_2, p_3 и манипулируя значениями шагов во времени и пространстве, можно моделировать процесс диффузии с коэффициентом диффузии в широком диапазоне [4].

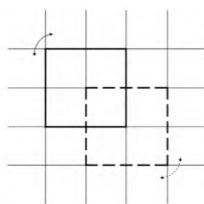


Рис. 1 – Фрагмент КА с окрестностью Марголуса

В нашей работе для построения эффективной программной реализации блочных клеточных автоматов мы предлагаем воспользоваться тензорным подходом [5]. Данный подход ориентирован на использование многоядерных видеокарт с привлечением специальной рабочей среды (фреймворка) TensorFlow для распараллеливания программ. Суть подхода заключается в следующем: 1) состояние автомата представляется в виде тензора (многомерная матрица), переход автомата из одного состояния в другое осуществляется при помощи операций над тензорами.

В ходе построения программной реализации для блочной модели обнаружены некоторые сложности (ограничения) в применении тензорного подхода. Первая (не столь критична) – функционал TensorFlow не позволяет осуществлять операции с тензорами произвольного ранга: нас, в первую очередь, интересует операция свёртки тензора с заданным ядром, когда ядро может быть тензором произвольного ранга. в TensorFlow доступна свёртка с ядром того же ранга, что и заданный тензор (например, для тензора второго ранга – случай для двумерного клеточного автомата, ядро тоже должно быть второго ранга). Вторая (более критична) – вероятностный характер поведения модели требует генерации на каждом такте работы автомата ядра свёртки определенного вида, что значительно повышает сложность алгоритма и влияет на эффективность подхода в целом. Для преодоления указанных сложностей операции поворота чётных и нечётных блоков автомата были

реализованы нами в рамках имеющейся в TensorFlow возможности внедрения собственных операций над тензорами на уровне манипуляции основными структурами данных (массивами), которыми представляются тензоры. Библиотеки разработчиков TensorFlow позволяют реализовывать операции над данными как программы для центрального процессора, так и как программы для графических ускорителей на языке CUDA от NVIDIA.

Для верификации нашей реализации модели диффузии на основе блочного синхронного клеточного автомата с окрестностью Марголуса мы повторили вычислительный эксперимент, описанный в [6]. Эксперимент состоит в вычислении глобального состояния клеточного автомата размерами 1000 на 1000 клеток после 5000 тактов работы. В качестве начального глобального состояния выбирается ситуация, в которой всё вещество сконцентрировано в центральном квадрате размером 100 на 100 клеток (рис. 2). Качественное сравнение результатов (рис. 3) позволяет заключить, что наша реализация сопоставима по точности моделирования диффузии с известными аналогами. Расчёт эволюции модели для 5000 тактов заняло для нашей системы (процессор Intel Core i7-8700K, ОЗУ 24 Гб, видеокарта NVIDIA GeForce RTX 2080 Ti, ОЗУ видеокарты 11 Гб) 10 секунд. К сожалению, авторы работы [6] не указали время работы программы для их реализации, поэтому сравнить эффективность нашей работы на данный момент не представляется возможным.

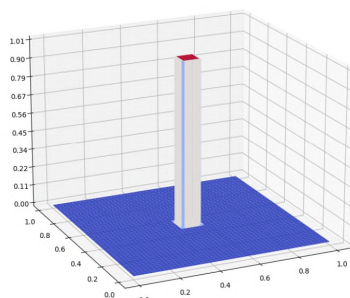
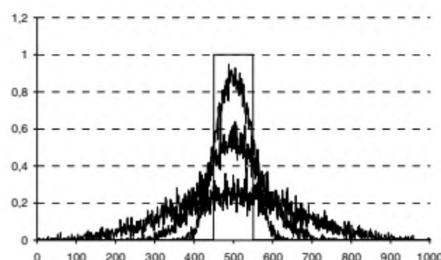
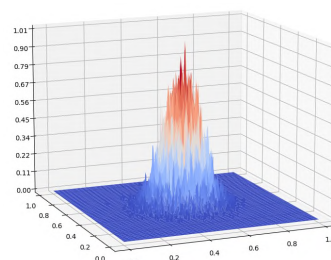


Рис. 2 – Начальное состояние автомата



а)



б)

Рис. 3 Результаты моделирования: а) из [6], б) полученные нами

В дальнейшем мы планируем провести эксперименты для выяснения эффективности нашей реализации по сравнению с известными реализациями клеточного автомата с окрестностью Марголуса для различных многопроцессорных систем.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 20-31-90070 Аспиранты.

Список публикаций:

- [1] Moore E. F. *Gedanken-experiments on sequential machines*, in C. Shannon and J. McCarthy ed. // *Automata Studies Princeton University* 1956. pp. 129–153.
- [2] Shalyapina N. A., Gromov M. L. «Life» in Tensor: Implementing Cellular Automata on Graphics Adapters // *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2019. № 31(3). pp. 217–228. [https://doi.org/10.15514/ISPRAS-2019-31\(3\)-17](https://doi.org/10.15514/ISPRAS-2019-31(3)-17).
- [3] Toffoli T., Margolis N. *Cellular automata machines: a new environment for modeling* // MIT Press, Cambridge, Massachusetts. 1987.
- [4] Бандман О. Л. *Клеточно автоматные модели пространственной динамики* // *Системная информатика*. 2006. № 10. С. 59–113.
- [5] Matolygin A. K., Shalyapina N. A., Gromov M. L., Torgaev S.N. *Tensor approach to software implementation of cellular automata model of diffusion* // *Journal of Physics: Conference Series*. 2020. Vol. 1680. pp. 1–6. doi:10.1088/1742-6596/1680/1/012035.
- [6] Medvedev Yu. G. *Multi-particle Cellular-Automata Models for Diffusion Simulation* // *Methods and Tools of Parallel Programming Multicomputers. MTPP 2010. Lecture Notes in Computer Science*, vol. 6083. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14822-4_23.