

# РАЗРАБОТКА ПРОТОТИПА ПАРАЛЛЕЛЬНОЙ ВЕРСИИ БИБЛИОТЕКИ КЛЕТОЧНО- АВТОМАТНЫХ ТОПОЛОГИЙ

Подготовила Трубицына Юлия, ФИТ, группа 19202

Руководитель: Медведев Юрий Геннадьевич старший

научный сотрудник ИВМиМГ СО РАН

Соруководитель: Киреев Сергей Евгеньевич

научный сотрудник ИВМиМГ СО РАН

# Введение

Ранее:

- Была проделана работа по реализации асинхронного клеточного автомата<sup>1</sup>, покрывающего поле максимальным количеством домино.
- Разработана ручная реализация синхронного клеточного автомата.
- Реализован синхронный клеточный автомат с помощью библиотеки клеточно-автоматных топологий.
- Разработана параллельная версия библиотеки, работающая на размерах поля кратных количеству процессов.

<sup>1</sup> Hoffmann, R., Désérable, D., Seredyński, F. A cellular automata rule placing a maximal number of dominoes

## Представление библиотеки

- Библиотека разрабатывается с целью сделать процесс разработки клеточно-автоматных моделей более удобным для прикладного программиста.
- Библиотекой подразумевается три программных модуля: препроцессор, симулятор и постпроцессор.
  - Препроцессор - производит инициализацию управляющих структур (например в данном модуле задается размер поля, размер клетки)
  - Симулятор – применяет функцию переходов ко всем клеткам модели заданное количество итераций.
  - Постпроцессор – отображает выходные данные

# Возможности библиотеки

	Последовательные ВС	Параллельные ВС с общей памятью	Параллельные ВС с распределенной памятью	Графические ускорители
Синхронный КА	есть	есть	цель этой работы	планируется в перспективе
Асинхронный КА	есть	разрабатывается в другом проекте	планируется в перспективе	планируется в перспективе

# Цели и задачи

Цель:

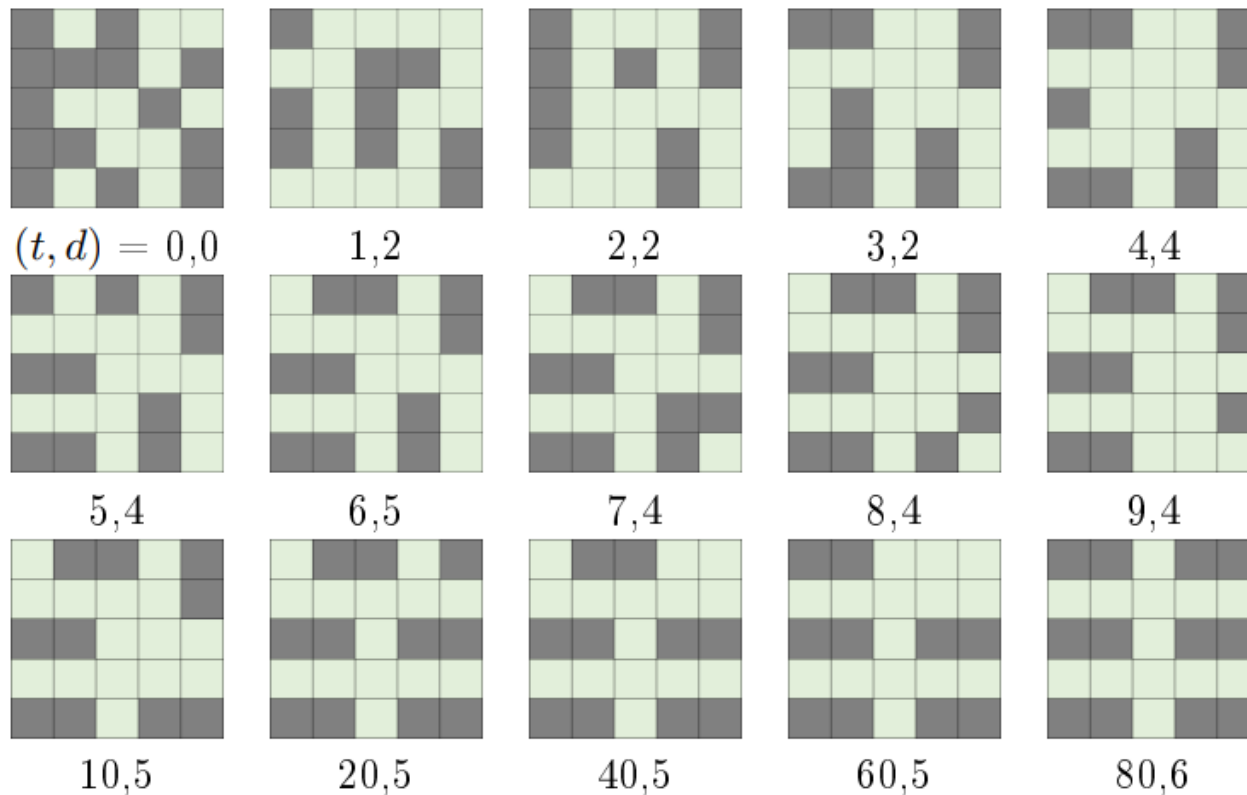
- Разработка версии библиотеки клеточно-автоматных топологий для параллельных вычислительных систем с распределенной памятью.

Задачи:

- Добавить возможность запускать параллельную версию библиотеки на произвольном количестве процессов.
- Проверить корректность работы библиотеки на модели “Домино”.
- Протестировать программу, написанную с помощью библиотеки, с целью измерения времени работы.

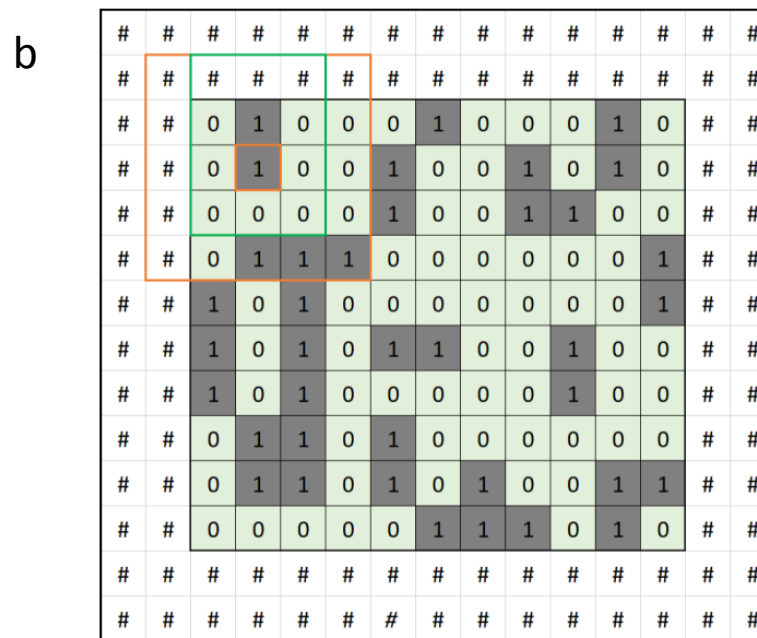
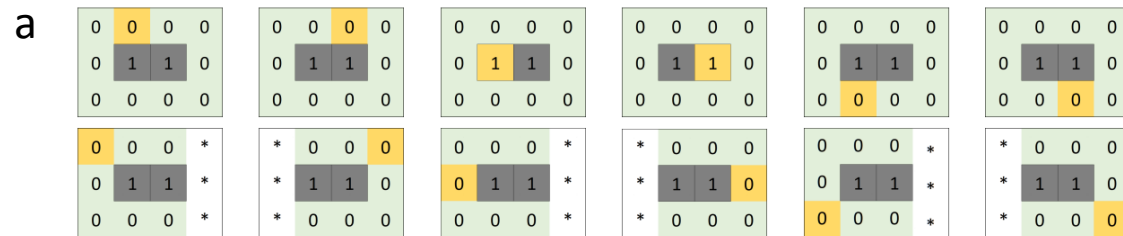
# Описание модели

- Входные данные представлены квадратным полем со случайным состоянием клеток.
- Возможные состояния клеток: "0", "1" и "#".



# Описание модели

1. Прикладываем к выбранной клетке и её соседям шаблоны, проверяем их совпадение. Правило прикладывания: выделенной клеткой шаблона на выбранную клетку клеточного массива. В сравнении не участвует выделенная клетка шаблона и клетки '\*'.



На рисунке:

- a) Шаблоны - представление плиток домино с метками (желтый квадрат).
- b) Прикладывание шаблона к выбранной клетке.

# Описание модели

2. В зависимости от того, сколько и каких шаблонов совпало, вычисляем значение  $h$ .
3. В зависимости от значения  $h$  вычисляем новое состояние выбранной клетки.

Функция переходов:

$$s'(i, j) = \begin{cases} \text{random} \in \{0, 1\} & \text{if } h = 0 \text{ with probability } \pi_0 \\ \text{random} \in \{0, 1\} & \text{if } h = 1 \text{ with probability } \pi_1 \\ 0 & \text{if } (h = 2) \text{ or } (h = 3) \text{ or } (h = 4) \\ 1 & \text{if } (h = 100) \\ s(i, j) & \text{otherwise} \end{cases} \quad (1)$$

$h$  – счетчик попаданий клетки в шаблоны



# Текущие результаты

## Способ распараллеливания

- Основная работа с полем происходит в симуляторе, из-за чего было решено разработать параллельные версии функций, которые он использует.
- Для распараллеливания был выбран метод декомпозиции области моделирования.
- С помощью библиотеки MPI была произведена 1D декомпозиция.

# Реализация параллельной версии для работы на произвольном количестве процессов

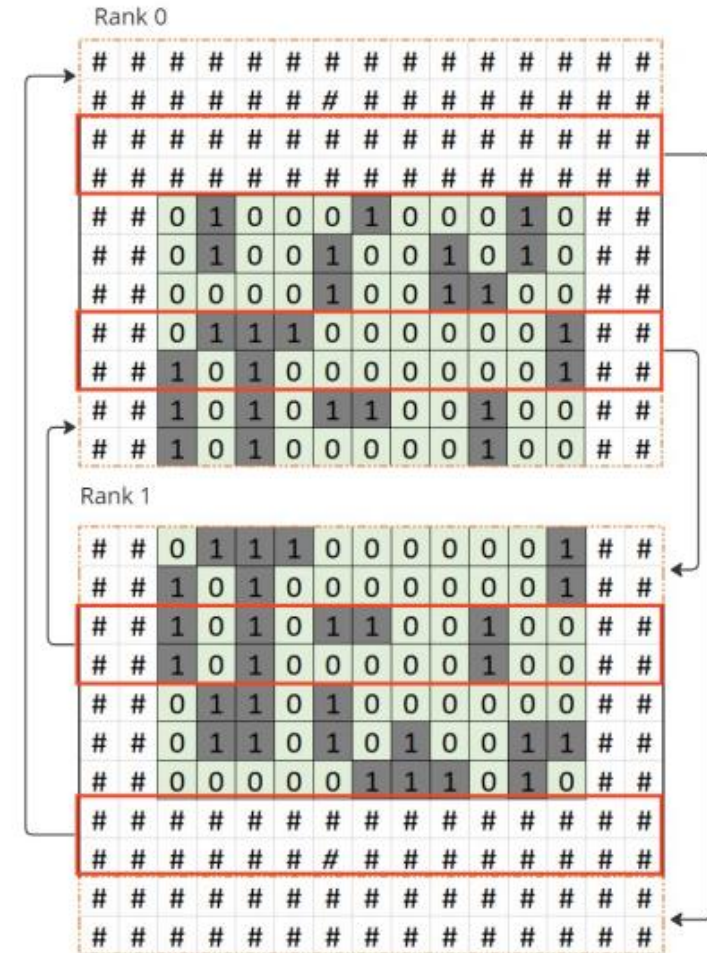
- **Функция инициализации симулятора**
  - Главный процесс считывает поле по кусочкам и передает их другим процессам

#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	0	1	0	0	0	1	0	0	0	1	0	#	#
#	#	0	1	0	0	1	0	0	1	0	1	0	#	#
#	#	0	0	0	0	1	0	0	1	1	0	0	#	#
#	#	0	1	1	1	0	0	0	0	0	0	1	#	#
#	#	1	0	1	0	0	0	0	0	0	0	1	#	#
#	#	1	0	1	0	1	1	0	0	1	0	0	#	#
#	#	1	0	1	0	0	0	0	0	1	0	0	#	#
#	#	0	1	1	0	1	0	0	0	0	0	0	#	#
#	#	0	1	1	0	1	0	1	0	0	1	1	#	#
#	#	0	0	0	0	0	1	1	1	0	1	0	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#

# Реализация параллельной версии для работы на произвольном количестве процессов

- **Функция итерации**

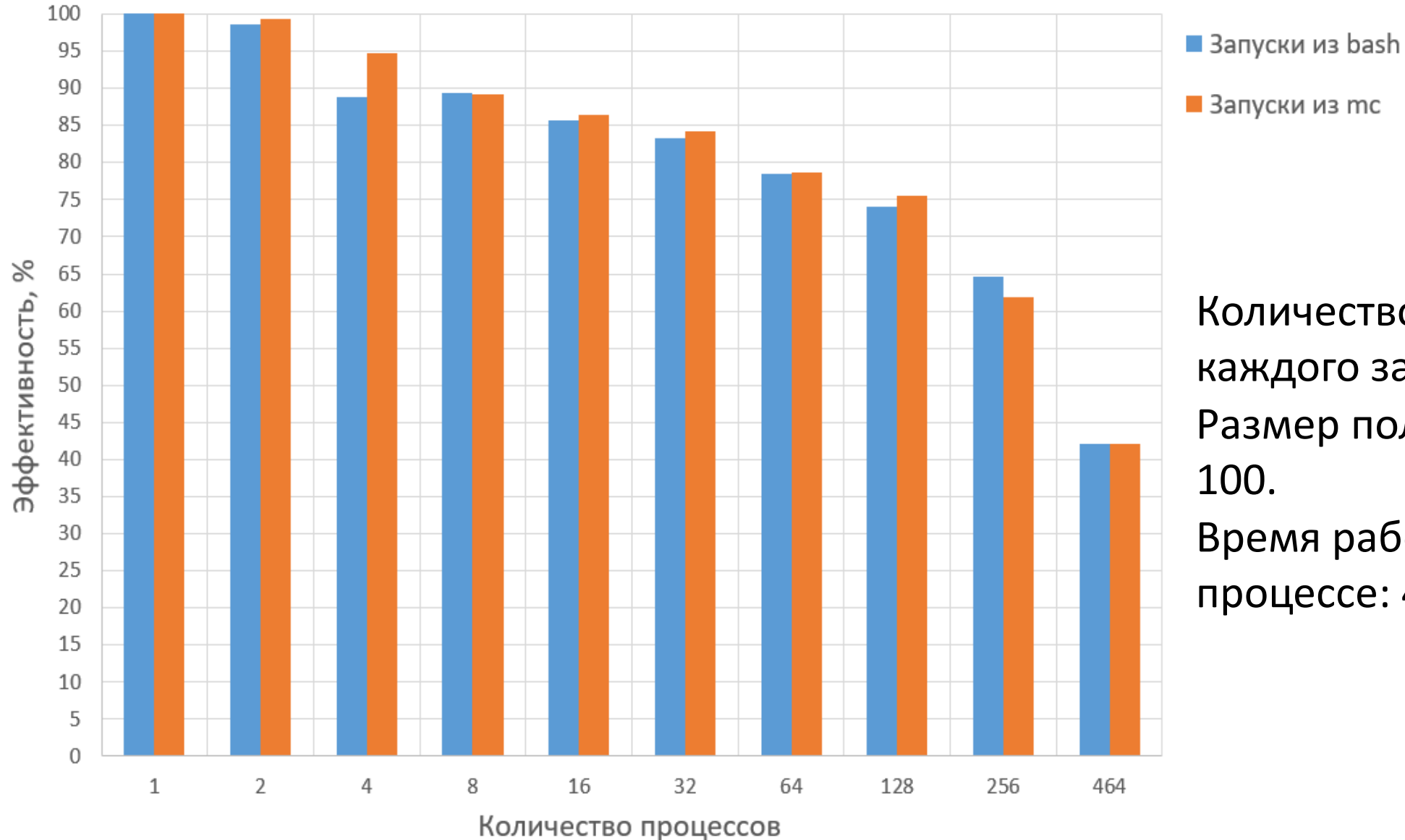
- Применяется для каждой части поля независимо.
- Для имитации синхронной работы автомата используется вспомогательный массив, куда записываются новые состояния клеток.
- После прохождения итерации и обновления состояний клеток части поля передают друг другу границы.



# Реализация параллельной версии для работы на произвольном количестве процессов

- Функция финализации симулятора
  - Главный процесс собирает по очереди все части поля и записывает получившийся результат в файл.

# Тестирование



Количество итераций для каждого запуска:  $T = 1\ 000$ .  
Размер поля:  $I = 10000$ ,  $J = 100$ .  
Время работы на одном процессе: 484,69 секунды

# Результаты

На зимней школе удалось выполнить все поставленные задачи:

- Добавлена возможность запускать параллельную версию библиотеки на произвольном количестве процессов.
- Была проверена корректность работы библиотеки на модели “Домино”.
- Оценена эффективность параллельной реализации клеточного автомата “Домино”.

# Планы

- В симуляторе организовать режим обмена границами на фоне вычислений внутренних строк частей клеточного массива, обрабатываемых процессами.
- В дальнейшем планируется счет больших задач для заказчиков и совместная статья с ними.



**СПАСИБО ЗА ВНИМАНИЕ!**