

# Летняя школа по параллельному программированию 2012

**Название проекта:**

Клеточно-автоматное моделирование синхронного  
режима разделения фаз с помощью MPI

**Исполнитель проекта:**

Свичкарев Анатолий,  
1 курс ФИТ НГУ

**Руководитель проекта:**

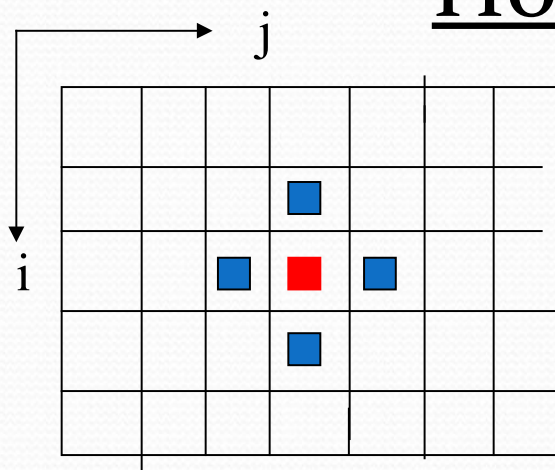
Шарифулина Анастасия

**13 июля 2012**

## План доклада:

1. Постановка задачи
2. Идея решения
3. Реализация
4. Визуализация
5. Тестирование
6. Заключение

# Постановка задачи

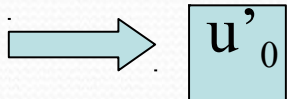
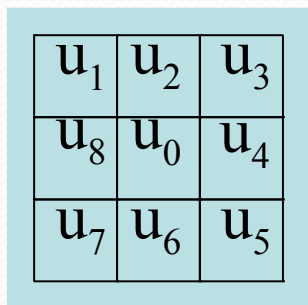


$A = \{0, 1\}$  - алфавит  
 $X = \{(i, j) : i, j = 0, \dots, N\}$

**Синхронный режим**

$$u'_i = \begin{cases} 1, & \text{if } \sum u_k > 5 \text{ or } \sum u_k = 4 \\ 0, & \text{if } \sum u_k < 4 \text{ or } \sum u_k = 5 \end{cases}$$

**Тоталистический режим**

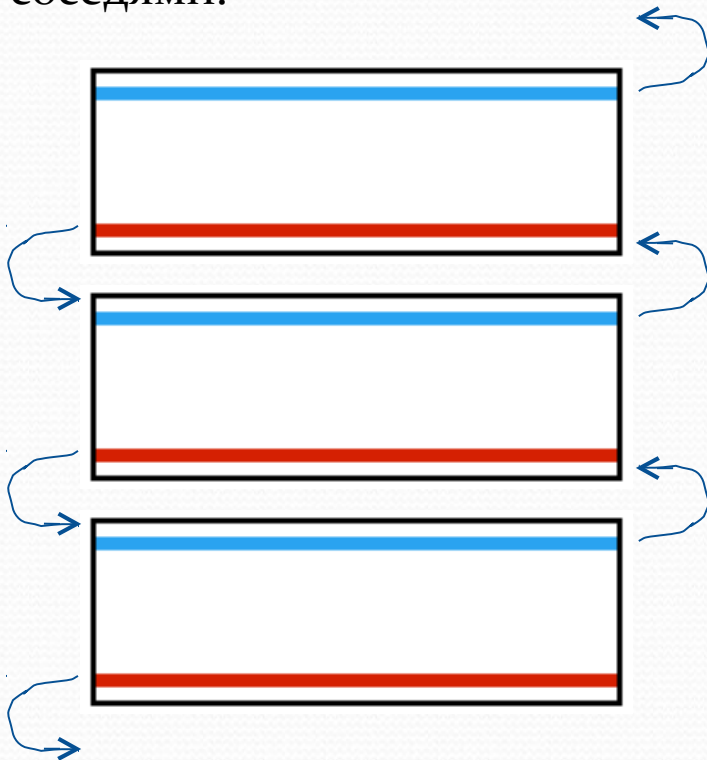


*Любой порядок применения  
 локального оператора  
 одновременная смена состояний*

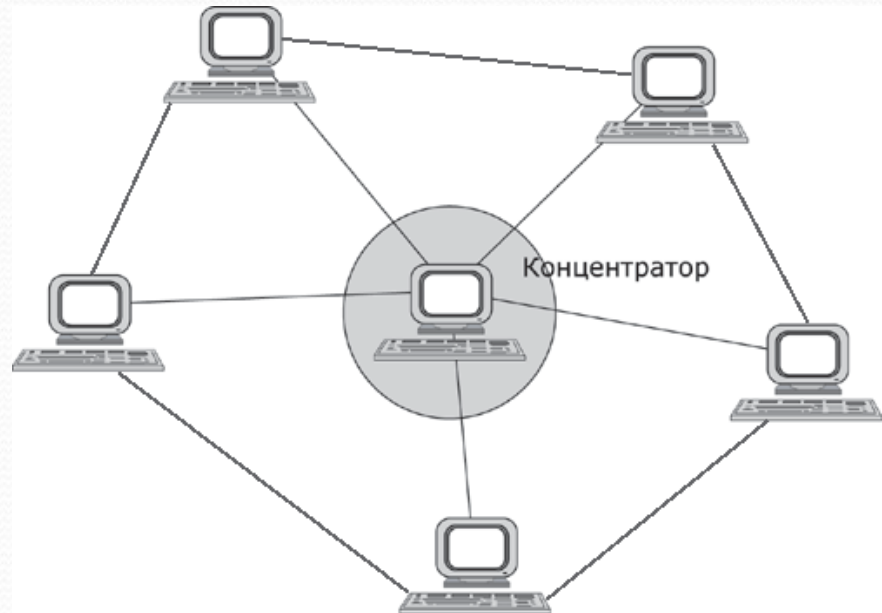
# Идея решения

## Декомпозиция:

Разбивка исходного массива с данными на полосы равной длины, которые посылаются отдельному процессу для вычислений. После каждой итерации процессы обмениваются граничными полосами со своими соседями.



Была применена модернизированная топология «Звезда»



# Реализация

Для распараллеливания была использован программный интерфейс MPI. Концентратором является процесс с нулевым рангом, остальные (size-1) процессов занимались обработкой каждый своего розданного блока.

В каждой итерации процесс вначале проводил подсчёт своих граничных полос данных, сразу асинхронно неблокированно их отправлял соседям и инициализировал так же приём данных от соседей необходимых для следующей итерации границ данных.

В это время процесс проводил подсчёт своего внутреннего блока данных, независимого от соседей. Итерация заканчивалась функциями завершения неблокируемых операций.

Через указанное количество итераций результат выводится в файл с расширением .bmr.

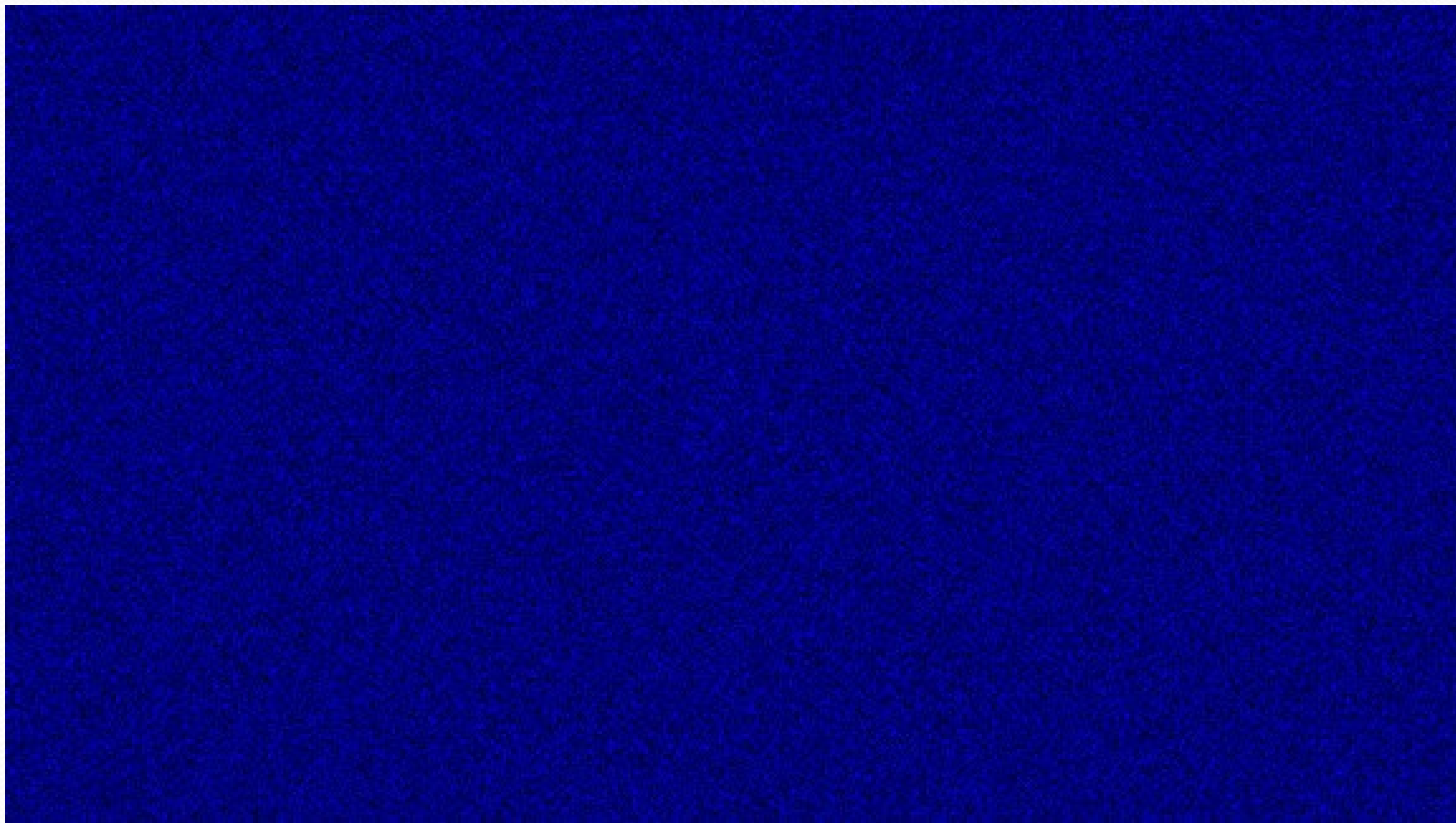
# Реализация

## **Основные ограничения:**

1) Запуск программы возможен минимум на 2 процессах, т.к. один процесс всегда является концентратором и в вычислениях не участвует.

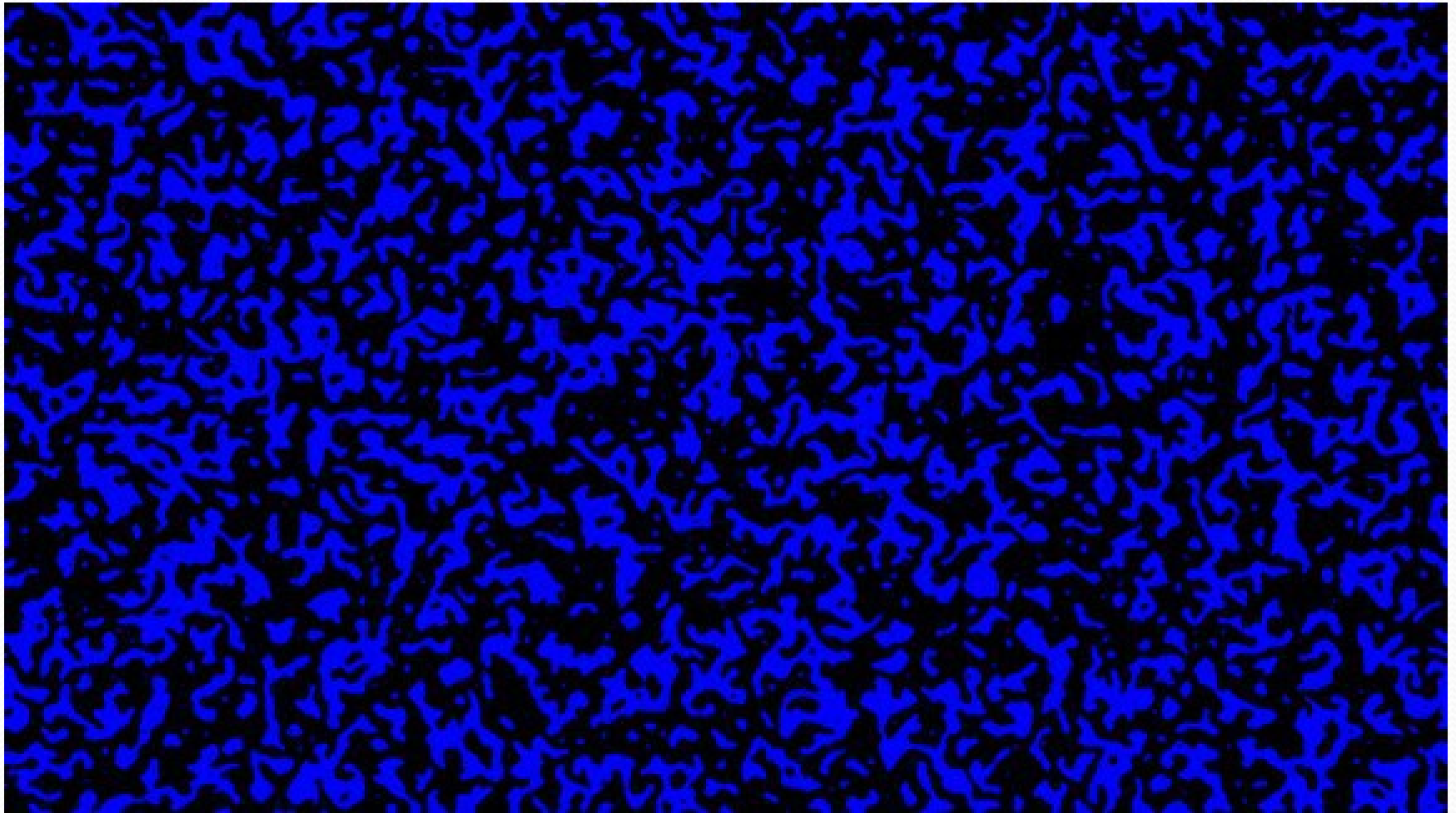
2) Количество строк должно быть кратно количеству процессов без концентратора, иначе не будут обрабатываться последние несколько строк.

# Визуализация



Начальное состояние КА

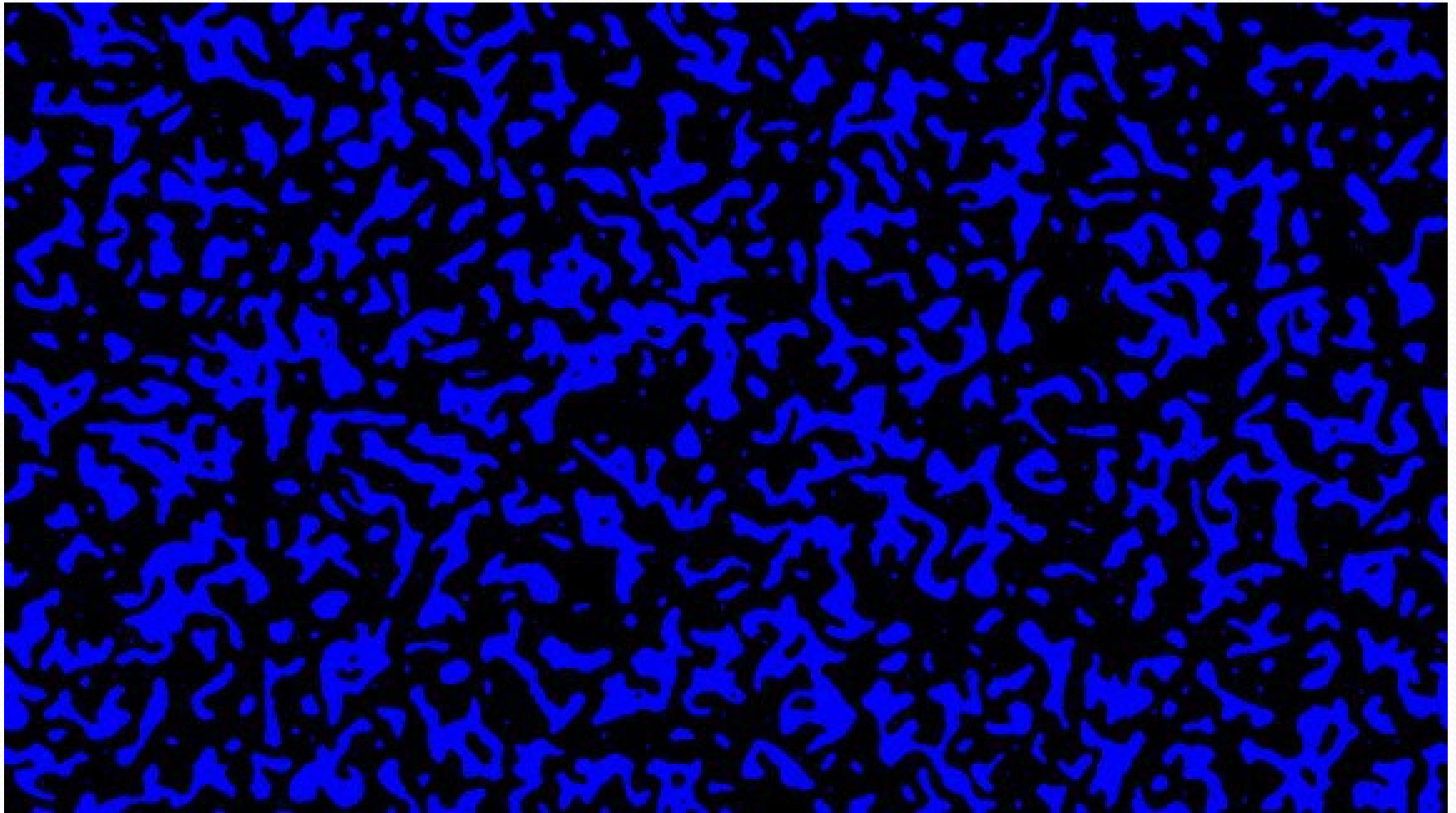
# Визуализация



Состояние КА через 100 итераций

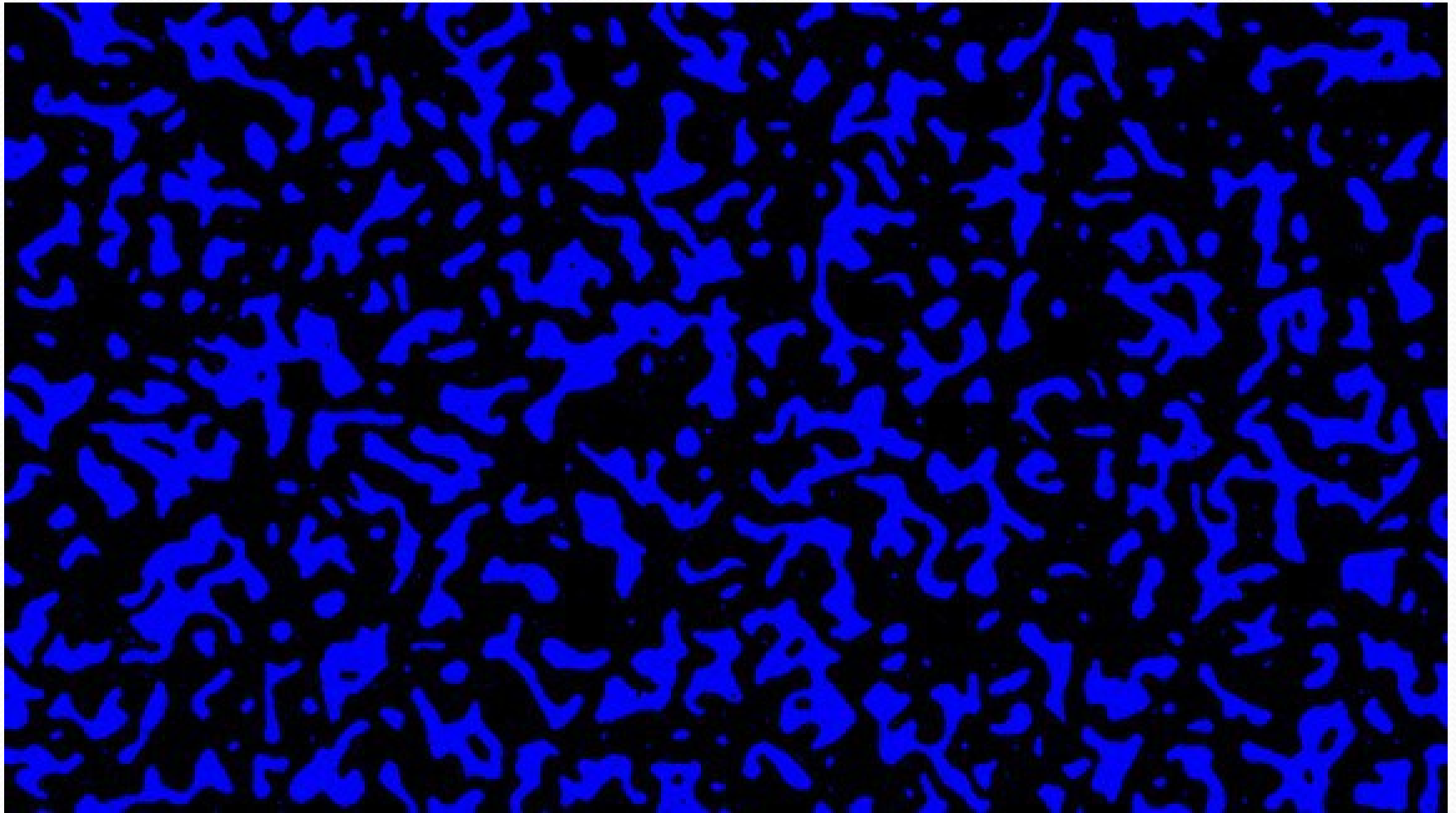


# Визуализация



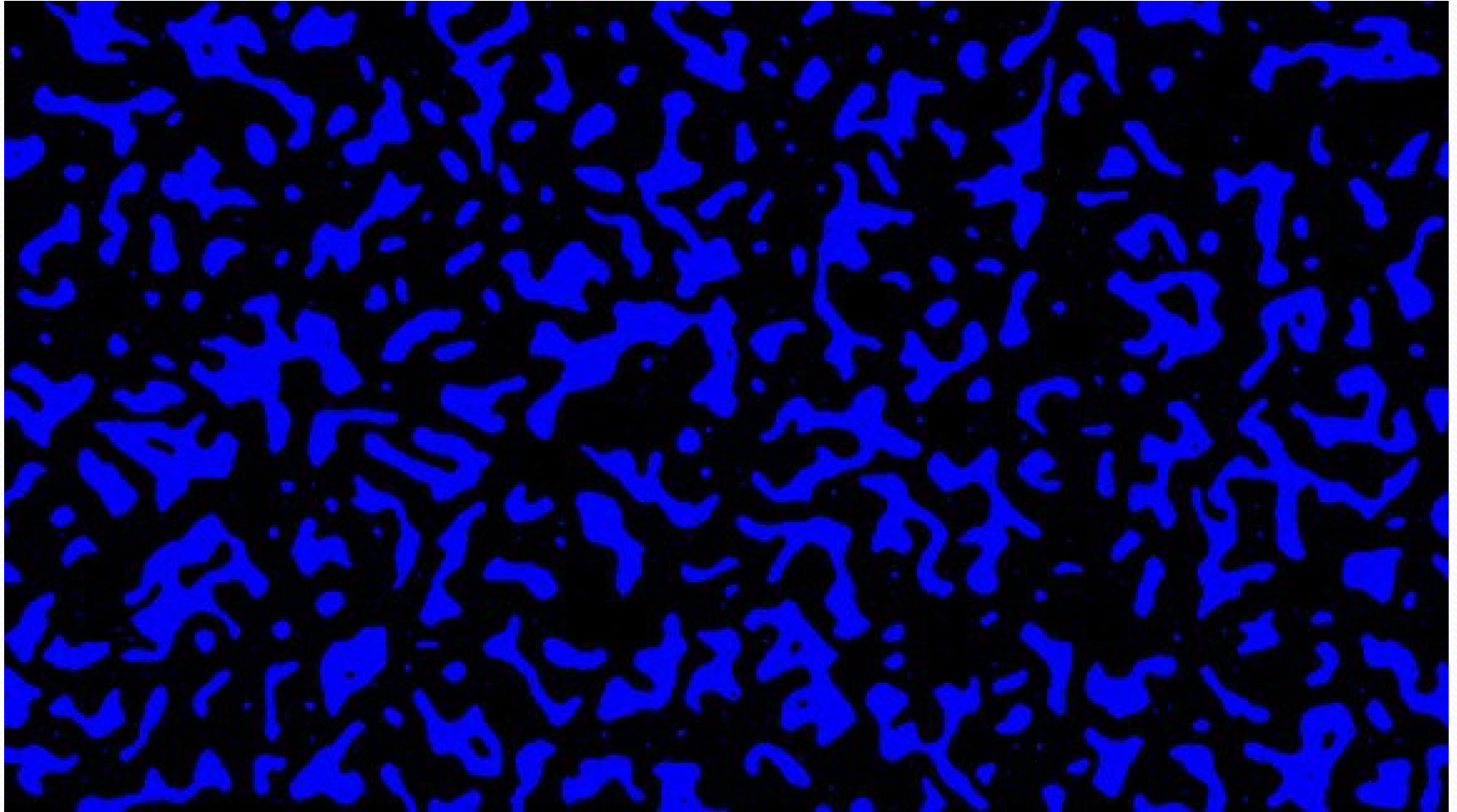
Состояние КА через 200 итераций

# Визуализация



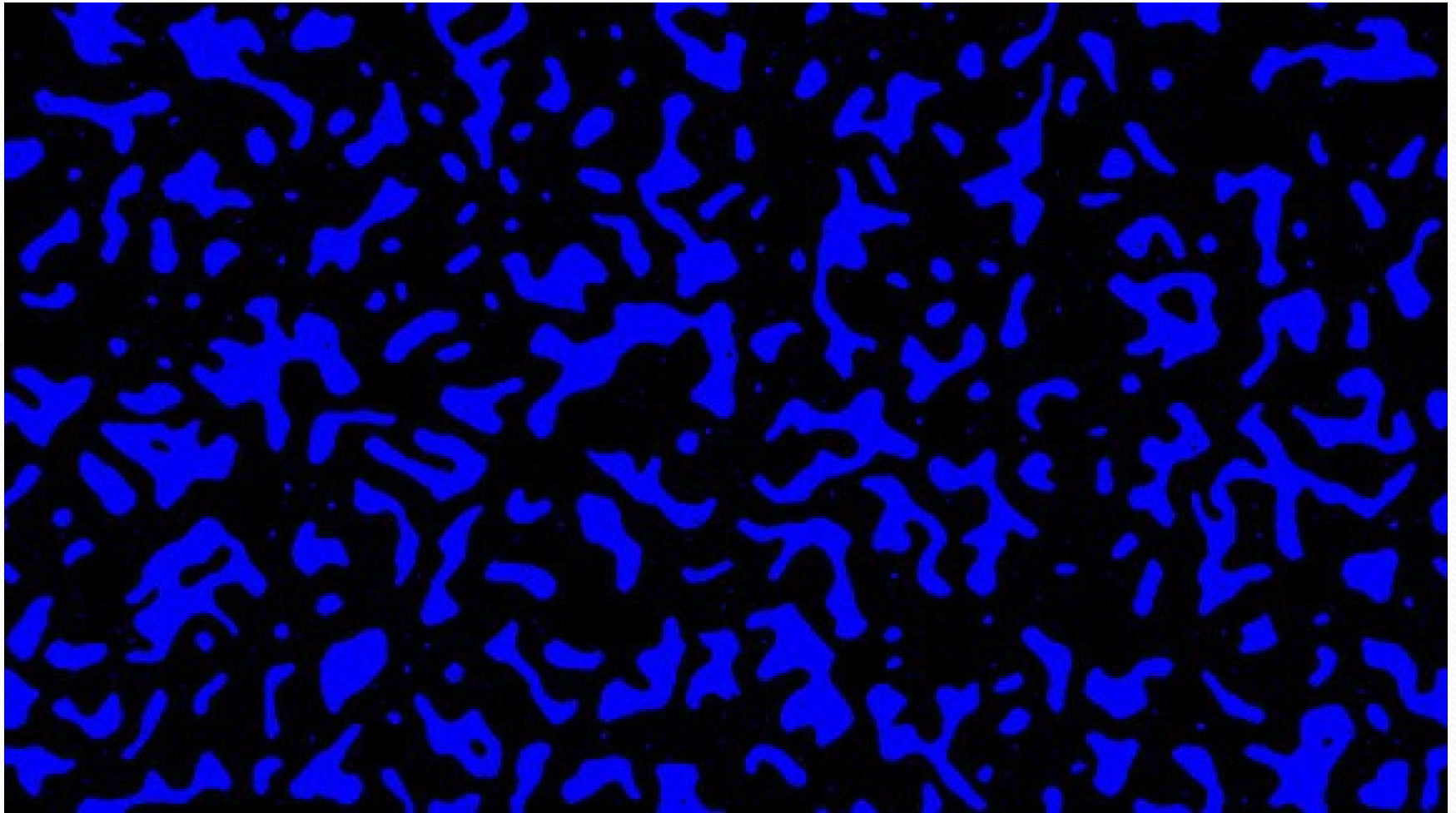
Состояние КА через 300 итераций

# Визуализация



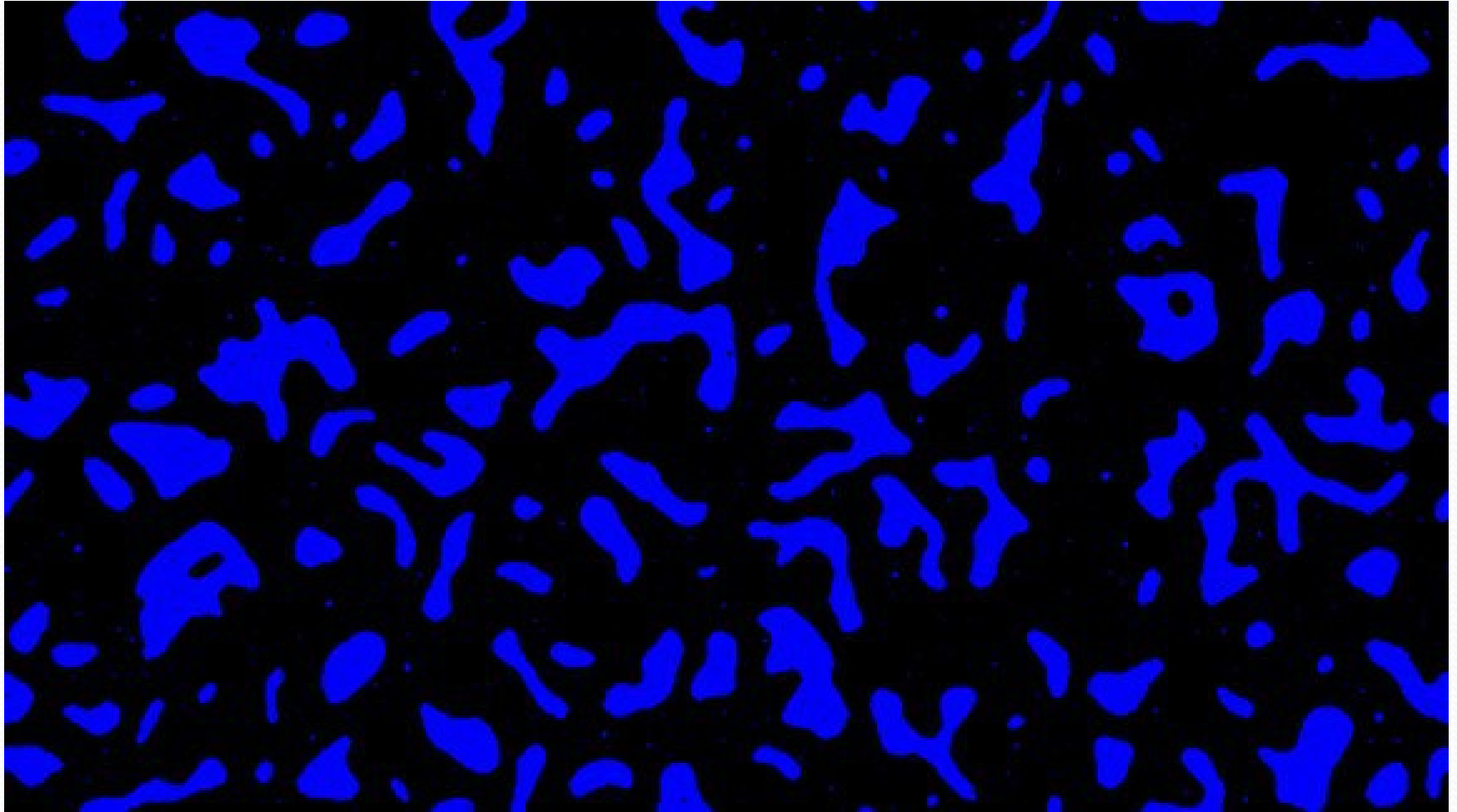
Состояние КА через 400 итераций

# Визуализация



Состояние КА через 500 итераций

# Визуализация



Состояние КА через 1000 итераций

# Тестирование

## Тесты проводились по трём критериям:

- Среднее время на итерацию (  $T(np)$  ) в миллисекундах (по 10 итераций запускалось)

- Ускорение

$$S(np) = \frac{T_1}{T(np)}$$

где  $T_1$  – время работы при запуске на одном процессе ( в моём случае на двух

- Эффективность

$$Q(np) = \frac{S(np)}{np}$$

# Тестирование

**При размерности матрицы 3000 x 3000**

np	2	3	4	5	6	7
T(np)	990	512	353	255	312	300
S(np)	1,00	1,89	2,80	3,88	3,17	3,00
Q(np)	1,00	0,94	0,93	0,96	0,55	0,50

**При размерности матрицы 6000 x 6000**

np	2	3	4	5	6	7
T(np)	4000	2000	1340	1010	850	820
S(np)	1,00	2,00	2,97	3,96	4,71	4,87
Q(np)	1,00	0,97	0,99	0,99	0,94	0,81

# Заключение

## **Основных результатов работы:**

- Изучена модель клеточного автомата на примере разделения фаз;
- Освоена работа с моделью параллельных вычислений на MPI;
- Создана однопоточная версия с использованием OpenGL;
- Создана параллельная версия программы с возможностью сохранения результатов в графическом виде;
- Было проведено тестирование на ускорение и эффективность программ на разных количествах процессов.



# Заключение

## **Возможных направлений развития проекта :**

- Создать ввод начального состояния из картинки;
- Создать возможность моделирования любых других правил моделей КА;
- Создать полноценную топологию на MPI.



Спасибо за внимание!