Отчет о проделанной работе

Тема работы:

Методика совместного использования технологий MPI и OpenMP в библиотеке CATlib для двумерных топологий в синхронном режиме.

Новосибирск, 2025

Содержание

1. Аннотация

Отчёт посвящён разработке методики совместного использования технологий MPI и OpenMP в библиотеке CATlib для параллельного моделирования двумерных клеточных автоматов в синхронном режиме. Цель — реализация гибридного алгоритма, сочетающего крупноблочный (MPI) и мелкозернистый (OpenMP) параллелизм для повышения производительности вычислений.

Обоснована актуальность подхода: моделирование клеточных автоматов требует значительных ресурсов, а совмещённый параллелизм обеспечивает лучшее масштабирование на кластерах. Разработан алгоритм, использующий MPI для распределения доменов и обмена граничными данными и OpenMP для параллельной обработки клеток внутри процессов. Поддерживаются произвольные двумерные топологии.

Проведено тестирование на кластере ИВЦ НГУ с оценкой ускорения и эффективности по сценариям сильного и слабого масштабирования. Для модели «Игра жизнь» гибридный подход показал лучшие результаты по сравнению с чистыми MPI и OpenMP, особенно при числе ядер свыше 24. Для модели FHP-MP гибридный подход уступал чистому MPI, но показал потенциал при определенных конфигурациях.

Результаты подтверждают целесообразность гибридного подхода и дают направление для дальнейшей оптимизации. Работа представлена на 63-й Международной студенческой конференции в НГУ.

2. Состав коллектива

- Фомин Александр Тимофеевич – студент группы 21205, Факультет информационных технологий, Кафедра параллельных вычислений, Новосибирский национальный исследовательский государственный университет, Направление подготовки: 09.03.01 Информатика и вычислительная техника, Направленность (профиль): Программная инженерия и компьютерные науки.
- Медведев Юрий Геннадьевич. – руководитель ВКР, к.т.н. доц., каф. ПВ ФИТ НГУ, с.н.с. ИВМиМГ СО РАН.
- Киреев Сергей Евгеньевич – соруководитель ВКР, ст. преп., каф. ПВ ФИТ НГУ, н.с. ИВМиМГ СО РАН.
- Малышкин Виктор Эммануилович – заведующий кафедрой, д.т.н., профессор, г. н.с. ИВМиМГ СО РАН.

3. Научное содержание работы

3.1. Постановка задачи

Целью работы является разработка и внедрение гибридного параллельного алгоритма, использующего MPI и OpenMP, в библиотеку CATlib для синхронного моделирования двумерных клеточных автоматов. Задача состоит в развитии методики синхронного моделирования клеточных автоматов с использованием параллельных вычислений путем интеграции технологий MPI и OpenMP на двумерных топологиях. Предполагается, что такая интеграция **повысит производительность моделирования** за счет оптимального использования вычислительных ресурсов. Для достижения поставленной цели были решены следующие задачи:

1. Разработка алгоритма, обеспечивающего **совместное использование технологий MPI и OpenMP в библиотеке CATlib**, с фокусировкой на корректном распределении вычислений и взаимодействии между уровнями параллелизма.
2. Реализация данного алгоритма для двумерных топологий, включая уже существующие в библиотеке.
3. Проведение тестирования и сравнения производительности параллельных реализаций: только с использованием MPI, только с OpenMP, а также их комбинации. В качестве тестовых задач были выбраны модель «Игра жизнь» и модель решёточного газа FHP-MP.

3.2. Современное состояние проблемы

Современные вычислительные технологии, включая клеточные автоматы (КА), активно применяются для решения сложных задач в науке и технике.

Моделирование реальных процессов с использованием КА требует обработки больших объёмов данных и выполнения многочисленных итераций, что делает такие задачи масштабными и ресурсоёмкими. Для повышения производительности моделей КА требуется **эффективное распараллеливание вычислений**.

Существует актуальность в разработке и исследовании методов **совмещённого параллелизма** (крупноблочного и мелкозернистого) в алгоритмах клеточных автоматов для более эффективной масштабируемости на современных кластерных

вычислительных системах. Несмотря на успешные решения в отдельных прикладных областях, вопросы **разработки и внедрения методов эффективной параллельной реализации КА-алгоритмов с использованием технологий MPI и OpenMP в рамках совмещенного подхода исследованы недостаточно полно**.

Существующие программные инструменты для моделирования КА часто имеют ограничения, такие как отсутствие поддержки крупномасштабных вычислений, пользовательских топологий или эффективных параллельных алгоритмов. Библиотека CATlib разрабатывается для решения этих проблем, поддерживая интеграцию MPI и OpenMP.

3.3. Подробное описание работы, включая используемые алгоритмы

Работа посвящена разработке методики совместного использования технологий MPI и OpenMP в библиотеке CATlib для синхронного моделирования двумерных клеточных автоматов. Цель состояла в разработке и внедрении гибридного параллельного алгоритма, использующего MPI и OpenMP, в библиотеку CATlib для синхронного моделирования двумерных клеточных автоматов. Этот подход имеет потенциал для обеспечения более эффективной масштабируемости на современных кластерных вычислительных системах и может быть адаптирован к задачам пространственной динамики в дискретных моделях.

В работе разработан **гибридный параллельный алгоритм**, сочетающий возможности двух технологий: MPI (Message Passing Interface) для распределения вычислительной нагрузки между процессами и OpenMP (Open Multi-Processing) для распараллеливания вычислений внутри одного процесса. Совместное использование этих технологий направлено на эффективное задействование ресурсов вычислительной системы и достижение максимального ускорения моделирования клеточных автоматов.

В режиме распределённых вычислений используется MPI для взаимодействия между процессами. При разбиении клеточного массива на подрешётки, каждая из которых обрабатывается отдельным MPI-процессом, возникает необходимость в обмене граничными данными между соседними процессами. Это обусловлено тем, что состояние каждой клетки зависит от состояния её соседей, которые могут находиться в подрешётках других процессов.

На каждой итерации моделирования выполняется обмен значениями на границах, чтобы обеспечить корректное применение функции перехода к граничным клеткам. Для минимизации времени простоя и лучшего перекрытия вычислений с передачей данных используется неблокирующий обмен с помощью вызовов `MPI_Isend` и `MPI_Irecv`.

Последовательность вычислений организована следующим образом: сначала функция перехода применяется к внутренним клеткам подрешётки, состояние которых не зависит от данных соседних процессов. Это позволяет начать вычисления сразу после инициализации обмена. После завершения передачи данных (контролируется вызовом `MPI_Waitall`) производится вычисление значений для граничных клеток. Использование явной барьерной синхронизации (`MPI_Barrier`) не требуется, поскольку `MPI_Waitall` обеспечивает необходимую координацию между процессами.

На первой итерации обмен граничными значениями не выполняется, так как после чтения входного файла и распределения данных каждый процесс уже располагает актуальными значениями на своих границах.

На уровне отдельного процесса обработка данных внутри каждого MPI-процесса выполняется с использованием OpenMP. Благодаря синхронному режиму работы симулятора обновление состояний всех клеток происходит одновременно, что позволяет безопасно распараллелить обработку без гонок данных.

Основной задачей является применение пользовательской функции перехода клеточного автомата к каждой клетке сетки. Для распараллеливания используется набор директив OpenMP: `#pragma omp parallel` создаёт параллельную секцию, а `#pragma omp for` используется для распараллеливания основного цикла по клеткам. В случае вложенных циклов применяется директива `collapse`, позволяющая объединить несколько уровней вложенности и эффективнее распределить работу между потоками.

Стратегия распределения итераций между потоками может задаваться с помощью параметра `schedule(...)`. Например, стратегия `runtime` позволяет

выбрать конкретную схему через переменную окружения `OMP_SCHEDULE`, что удобно для экспериментов и настройки производительности без перекомпиляции кода.

Каждому потоку выделяется собственный буфер, в который загружаются данные о соседних клетках. Затем вызывается пользовательская функция перехода состояний, вычисляющая новое состояние клетки на основе данных о ее соседях. Результаты сохраняются во временный массив, после чего указатели на массивы текущего и следующего состояний обмениваются местами.

Библиотека поддерживает несколько типов топологий, каждая из которых требует особого подхода к обмену граничными данными и определению соседей.

SQUARE4 и SQUARE8. Для квадратных топологий с окрестностями фон Неймана (SQUARE4) и Мура (SQUARE8) используется стандартная схема обмена граничными данными между процессами. При этом передаются только ближайшие строки клеточного массива — следующая и предыдущая. Это позволяет эффективно поддерживать корректное моделирование без дополнительной обработки.

HEXAGON. Гексагональная топология требует более сложного подхода из-за особенностей сетки. В данной реализации используется смещенная квадратная сетка, в которой чётные и нечётные строки имеют разные схемы расположения соседей. При вычислении соседей необходимо учитывать чётность строки, что приводит к усложнению индексации. Дополнительная трудность возникает в случае использования MPI: каждый процесс создает собственный буфер с измененной нумерацией строк, что может нарушать целостность данных при обмене. Для решения этой проблемы реализовано распределение клеточного массива между процессами с учетом периодичности топологии, таким образом, чтобы строки, образующие один полный цикл смещения, находились в рамках одного MPI-процесса. Это позволяет выполнять локальные вычисления без глобальной перенумерации. Размер части массива, выделяемый каждому процессу, рассчитывается функцией `CAT_GetPartSize`, принимая во внимание кратность по числу строк. Недостатком такого подхода является то, что при перераспределении нагрузки необходимо передавать целые группы строк (т.е. один или несколько периодов), что увеличивает объём межпроцессных коммуникаций.

SQUARE24. Эта топология характеризуется расширенным радиусом взаимодействия (до второго порядка включительно), что означает наличие дальних соседей. Для корректной работы с такой топологией при обновлении состояний граничных клеток требуется передавать не только ближайшие строки, но и дополнительные строки, содержащие данные о более удалённых клетках. Глубина обмена (ширина граничной зоны) напрямую зависит от радиуса взаимодействия. В библиотеке предусмотрена возможность явного указания этой ширины с помощью директив препроцессора, например, `#define BORDER_SIZE 2`. Также реализована функция `CAT_GetBorderSize`, автоматически рассчитывающая размер граничной области по каждому измерению, что позволяет адаптировать алгоритм к произвольным радиусам взаимодействия.

Таким образом, поддержка различных топологий в библиотеке сопровождается адаптацией как локальных вычислений, так и схем обмена между MPI-процессами.

Для оценки эффективности разработанного гибридного параллельного алгоритма было проведено тестирование на кластерной системе ИВЦ НГУ.

Эксперименты проводились на кластере ИВЦ НГУ, доступном с тремя типами вычислительных узлов: bl2x220g7q (12 ядер/24 ГБ ОЗУ, до 6 узлов), xl230g9q (24 ядра/192 ГБ ОЗУ, до 3 узлов) и dl560g10q (80 ядер/384 ГБ ОЗУ, 1 узел).

Метрики производительности:

1. **Ускорение (Speedup, Sp):** Рассчитывается как отношение времени выполнения последовательной версии программы (Ts) ко времени параллельной версии (Tp) при использовании p процессов или потоков (Sp = Ts / Tp). Идеальное ускорение равно числу используемых ресурсов (Sp = p), на практике идеала достичь сложно из-за накладных расходов.
2. **Эффективность (Efficiency, Ep):** Показывает, насколько рационально используются ресурсы, как доля идеального ускорения (Ep = Sp / Np * 100%), где Np — количество используемых процессов или потоков. Высокая эффективность (близкая к 100%) свидетельствует об оптимальной реализации и масштабируемости. Значения ниже 70% указывают на значительные накладные расходы или недостатки.

Для оценки поведения алгоритма и влияния кэш-памяти использовались два подхода:

a. **Слабое масштабирование (weak scaling):** Размер задачи увеличивается пропорционально числу вычислительных ядер, сохраняя нагрузку на один процесс/поток примерно постоянной. Позволяет каждому потоку эффективно использовать кэш-память, снижает количество кэш-промахов. Эффективно для задач, сложность которых растет с числом потоков.
b. **Сильное масштабирование (strong scaling):** Размер задачи фиксирован, а число вычислительных ресурсов увеличивается. Позволяет оценить максимальное ускорение для ограниченного объема данных и измерить накладные расходы на параллелизацию. При увеличении числа потоков возрастает вероятность кэш-конфликтов и удаленных обращений к памяти (в NUMA-системах), что может снизить производительность.

Масштабируемость проверялась на различных конфигурациях ядер: $N \in \{1, 3, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72\}$.

Для чистой MPI-реализации тестировались все возможные комбинации распределения процессов по узлам. Для гибридной реализации варьировалось соотношение MPI-процессов и OpenMP-потоков. Для каждого N проверялись комбинации (n, p, t), где n — количество узлов, p — MPI-процессов на узел, t — OpenMP-потоков на процесс, при условии, что $n \times p \times t = N$, $n \leq 6$, $n \times p \neq 1$ и $t \neq 1$.

Выбор моделей:

a. **«Игра жизнь»:** Классическая модель, выбранная за простоту концепции и возможность построения сложных сценариев, что делает ее удобной для отладки и тестирования параллельных реализаций.
b. **Модель решеточного газа FHP-MP:** Выбрана для всесторонней оценки производительности на модели, отражающей характеристики реальных физических процессов и обладающей большей вычислительной сложностью.

Тестовые сценарии:

a. **Сильное масштабирование:**
    1. «Игра жизнь»: фиксированный размер массива 36635 × 36635 (около 1.25 ГБ), 1500 итераций, 10 запусков для каждой конфигурации.
    2. FHP-MP: фиксированный размер массива 26720 × 6250 (около 300 МБ), 100 итераций, 3 запуска для каждой конфигурации. Размер меньше из-за более сложной функции перехода и ограниченного времени.

b. **Слабое масштабирование:**

    1. «Игра жизнь»: размер массива увеличивался пропорционально числу ядер, базовый размер 14653 × 14653, 1500 итераций, 10 запусков.
    2. FHP-MP: тестировалось на наборе ядер $N \in \{1, 2, 4, 6, 8, 12, 14, 16, 18\}$, размер массива 2340 × 340 увеличивался пропорционально числу ядер, 150 итераций, 3 запуска. Тестирование проводилось в рамках одного вычислительного узла (проводилось на кластере НОЦ "Газпромнефть-НГУ").

3.4. Полученные результаты:

Модель «Игра жизнь»:

1. При сильном масштабировании OpenMP демонстрировал **почти линейный рост ускорения до 12–18 потоков**, затем производительность снижалась. MPI показывал хороший рост на малом числе процессов, но затем **ускорение резко снижалось** из-за роста коммуникаций. **Наилучшие результаты при сильном масштабировании демонстрировала гибридная реализация**, стабильно опережая чистые подходы при числе ядер от 24–30 и выше. Эффективность падала во всех подходах, но медленнее всего у гибридного.
2. При слабом масштабировании все модели показывали более сглаженное поведение. Гибридный подход оставался **наиболее устойчивым**, обеспечивая устойчивый рост ускорения за счет снижения межпроцессных сообщений и

эффективного использования многопоточности внутри узлов. Эффективность гибридной модели падала наименее выражено.

3. Результаты по модели «Игра жизнь» в целом совпали с предположениями, показав, что **гибридный подход превзошел MPI по ускорению и показал устойчивую масштабируемость**, достигая баланса между межпроцессорными и внутрипроцессными вычислениями.

Модель решеточных газов FHP-MP:

1. Результаты существенно отличались от модели «Игра жизнь». Гибридный подход **не продемонстрировал превосходства над чистой MPI-реализацией**; в ряде конфигураций показал худшие результаты. OpenMP показал низкое ускорение и быстро терял эффективность.
2. При сильном масштабировании **чистый MPI обеспечивал наиболее стабильный прирост производительности**. Гибридная модель не дала ожидаемого прироста; увеличение числа потоков в MPI-процессах снижало эффективность. Это могло быть связано с малым размером задачи и числом итераций, а также с вытянутой формой массива, влияющей на декомпозицию.
4. При слабом масштабировании на одном узле **чистый MPI демонстрировал почти линейный рост ускорения**, тогда как OpenMP насыщался раньше, а гибрид стартовал с наименьшим ускорением, но постепенно догонял OpenMP. Эффективность MPI оставалась высокой и устойчивой.
5. Наилучшие конфигурации гибридного режима часто характеризовались большим числом MPI-процессов на узел и меньшим числом OpenMP-потоков. В итоге, для модели FHP-MP в текущих условиях тестирования **чистый MPI остался наиболее надёжным методом**, но потенциал гибридного подхода при масштабировании задачи и подборе конфигураций отмечен.

5.1. Иллюстрации, визуализация результатов:

В работе приведены иллюстрации, демонстрирующие ускорение и эффективность различных подходов при сильном (Рис. 1, 3, 4) и слабом (Рис. 2, 5, 6) масштабировании для обеих моделей.

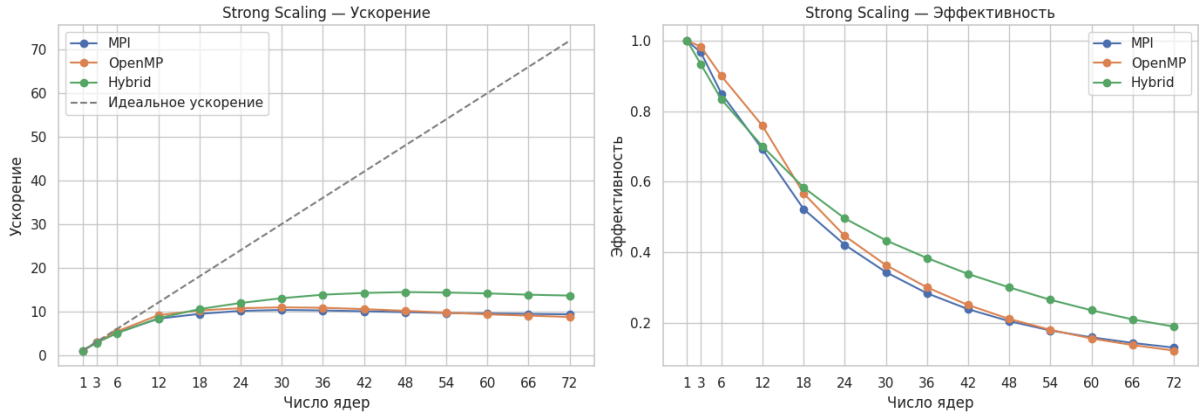

Рис. 1 - «Игра жизнь»: сильное масштабирование

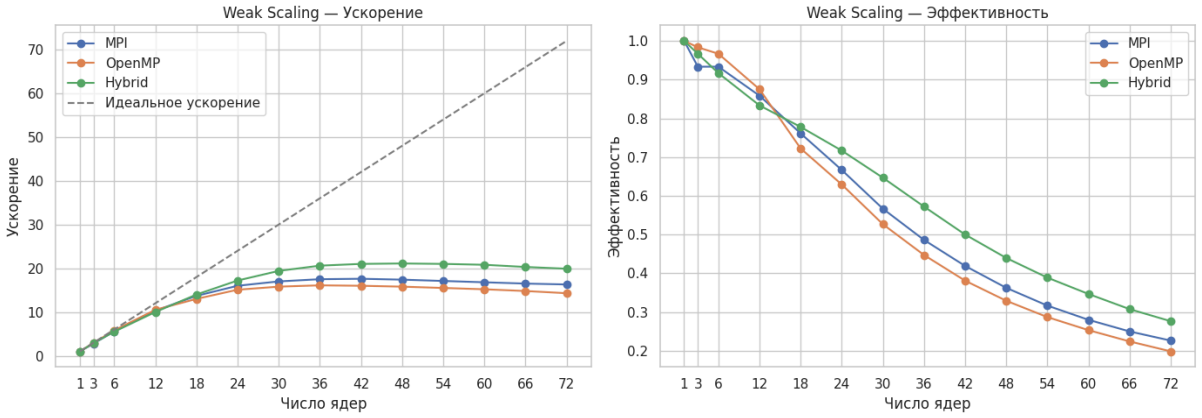

Рис. 2 - «Игра жизнь»: слабое масштабирование

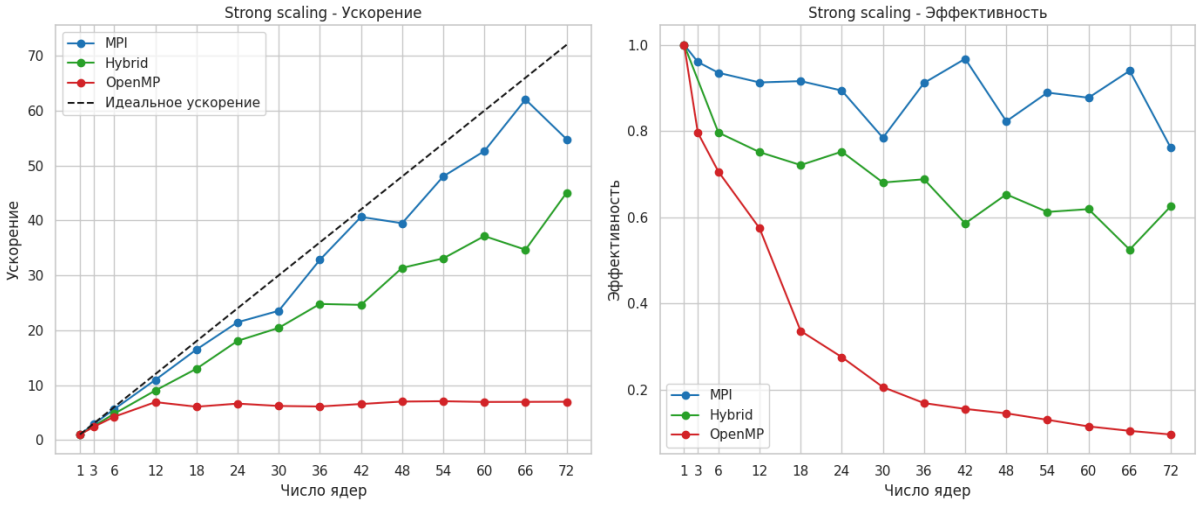

Рис. 3 - FHP-MP: сильное масштабирование

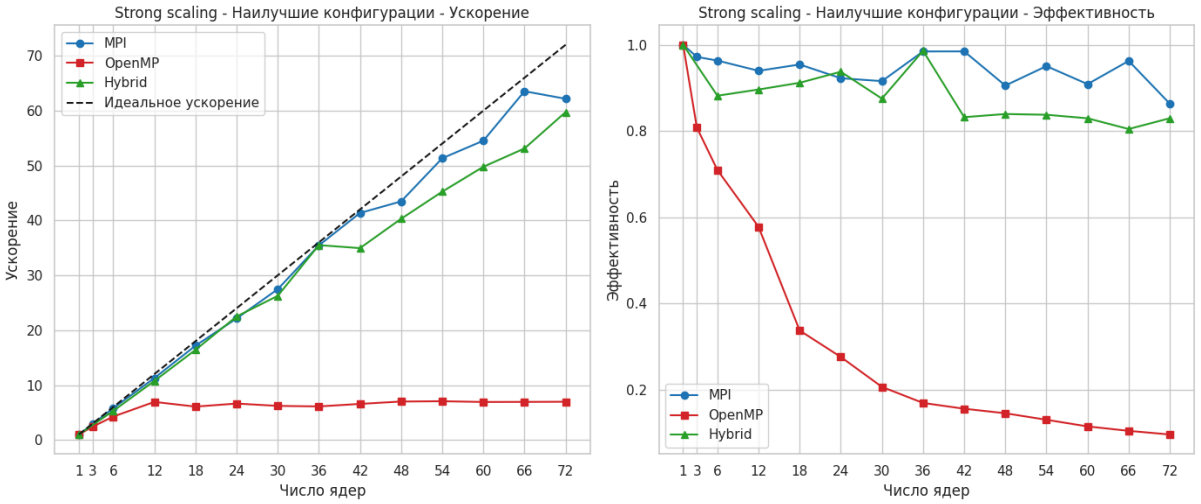

Рис. 4 - FHP-MP: сильное масштабирование лучшие конфигурации

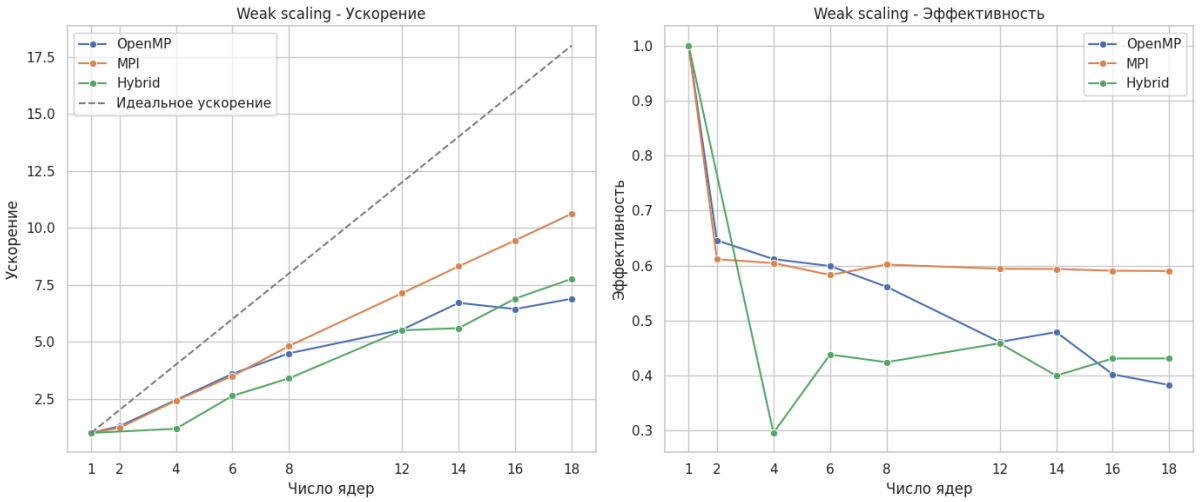

Рис. 5 - FHP-MP: слабое масштабирование

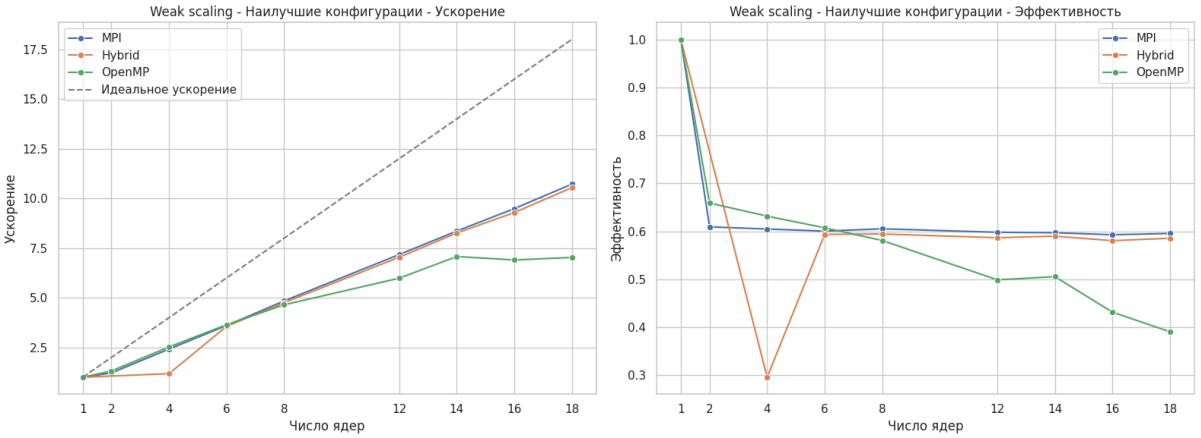

Рис. 6 - FHP-MP: слабое масштабирование лучшие конфигурации

6. Эффект от использования кластера в достижении целей работы

Использование кластерной системы ИВЦ НГУ позволило провести **широкомасштабные вычислительные эксперименты**, необходимые для **сравнения и оценки эффективности различных параллельных реализаций алгоритма**. Кластер предоставил доступ к различным конфигурациям вычислительных узлов, что дало возможность исследовать **масштабируемость разработанного гибридного алгоритма** и сравнить его с чистыми MPI и OpenMP подходами в режимах сильного и слабого масштабирования.

Проведенные тесты подтвердили **потенциал гибридного подхода** для модели «Игра жизнь» и выявили особенности его поведения для модели FHP-MP в заданных условиях. Результаты экспериментов, полученные на кластере, **являются основой для выводов о применимости гибридной модели** и определения перспективных направлений дальнейших исследований и оптимизации.

7. Перечень публикаций, содержащих результаты работы

Работа была представлена на **63-й Международной научной студенческой конференции** в НГУ с 16 по 22 апреля 2025 года в секции «Параллельные вычисления».

8. Впечатления от работы комплекса и деятельности ИВЦ НГУ, а также предложения по их совершенствованию

Впечатления от работы вычислительного комплекса ИВЦ НГУ положительные. Кластер предоставляет широкий набор вычислительных ресурсов и позволяет проводить масштабные параллельные эксперименты, включая как MPI-, так и OpenMP- и гибридные реализации. Работа с системой управления заданиями прошла без критических затруднений, вся инфраструктура производит впечатление зрелой и надежной.

Однако при тестировании параллельных реализаций возникли и некоторые сложности. Из-за высокой загруженности кластера задачи зачастую ожидали своей очереди в течение продолжительного времени, что растягивало полный цикл тестирования на несколько суток. Это особенно чувствительно при необходимости повторных прогонов для корректировки параметров или устранения ошибок.

Особое внимание хотелось бы обратить на поведение задач с некорректно запрошенными ресурсами. В текущей конфигурации такие задания продолжают оставаться в очереди, занимая место, но не будучи выполненными. Считаем, что система должна оперативно выявлять и снимать такие задачи с очереди, чтобы избежать потери времени и неэффективного использования ресурсов.

В целом, работа с кластером ИВЦ НГУ позволила получить ценный опыт в области высокопроизводительных вычислений и масштабирования параллельных алгоритмов. Хотелось бы выразить благодарность сотрудникам центра за предоставленные ресурсы и поддержку в ходе работы.