Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

На правах рукописи

Чой Е Рем

Параллельные алгоритмы матричного умножения и матричной экспоненты, основанные на асинхронных обменах данными между несколькими графическими ускорителями, и их применение для решения нестационарного уравнения Шредингера

РЕЗЮМЕ ДИССЕРТАЦИИ
на соискание учёной степени кандидата наук
по прикладной математике

Научный руководитель:
Доктор физико-математических наук,
профессор
Стегайлов Владимир Владимирович

Москва — 2024

Актуальность темы.

Развитие аппаратной базы суперкомпьютеров, в целом, опережает развитие параллельных алгоритмов, способных максимально эффективно решать задачи математического моделирования на новых типах вычислительного оборудования [1]. Одним из наиболее активно развивающихся типов высокопроизводительных серверов являются сервера с несколькими (4, 6, 8, 16) ГПУ-ускорителями, объединенными быстрыми каналами связи. Первый сервер такого типа DGX-1 с интерконнектом NVLink выпустила в 2016 году компания Nvidia, в настоящее время аналогичные сервера начинают выпускать компании AMD (с интерконнектом Infinity Fabric) и Intel (с интерконнектом Xe Link). Использование подобных серверов с несколькими ГПУ-ускорителями как единого вычислительного инструмента не является тривиальной задачей и требует разработки параллельных алгоритмов с оптимальной структурой обменов данными между ГПУ ускорителями. Подобная оптимальность может быть достигнута за счет наложения обменов данными и вычислений (принцип «overlapping computation and communication»). Актуальность диссертационной работы заключается в разработке подобных алгоритмов для матричного умножения и для расчета матричной экспоненты. Кроме того, в работе продемонстрировано, как на основе разработанных вычислительных алгоритмов можно использовать колоссальную вычислительную производительность серверов с несколькими ГПУ ускорителями для математического моделирования молекулярной динамики простейшего молекулярного иона H_2^+ на основе решения нестационарного уравнения Шредингера в формулировке с минимальным числом упрощающих предположений, что открывает возможность моделирования в до сих недостаточно изученной области неадиабатической молекулярной динамики.

Постановка проблемы

Целью работы является разработка параллельного алгоритма произведения матриц и матричной экспоненты на нескольких графических ускорителях, использующих высокопроизводительные каналы связи асинхронным образом для достижения максимальной производительности, и его применение для решения нестационарного уравнения Шредингера.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработка асинхронной параллельной программы эффективного алгоритма матричного произведения, выполняющей расчет на нескольких графических ускорителях, и исследование производительности данной программы.
2. Построение теоретической модели нахождения оптимального размера блоков, при которой исполнение программы будет выполнено с наилучшей производительностью.
3. Разработка и исследование программы алгоритма матричной экспоненты на графических ускорителях на базе использования разработанного алгоритма матричного произведения.
4. Решение нестационарного уравнения Шредингера методом использования ресурсов графических ускорителей высокой вычислительной мощности и анализ поведения волновой функции в изменяющемся от времени потенциале для молекулярного иона H_2^+.

Степень разработанности темы исследования

Обзор литературы показывает, что первым опубликованным алгоритмом матричного умножения, в котором была реализован принцип асинхронных обменов данными между ГПУ для улучшения производительности был алгоритм BLASX, описанный в статье Wang et al [2]. Несколько позже появился программный фреймворк PaRSEC [3], ориентированный на системы с процессорами IBM Power, имеющие быстрые каналы связи NVLink между ЦП и ГПУ, за счет чего задачу матричного умножения можно эффективно разделять между ЦП и ГПУ. Компания Nvidia предоставляет решение cuBLAS-XT с закрытым кодом для расчета матричного умножения одновременно на нескольких ГПУ и ЦП (именно cuBLAS-XT использован как референс в данной работе, и было показано, что предложенный алгоритм матричного умножения для нескольких ГПУ обеспечивает более высокую производительность). Одним из наиболее универсальных фреймворков для матричного умножения, объединяющим различные типы параллельных алгоритмов, является проект COSMA [4]. Методы разделения вычислений в COSMA основаны на оптимальных комбинациях размерностей матриц, количества процессоров и размеров памяти, при выборе которых за ключевую характеристику принимается оптимальность обеспечения операции чтения-записи за счет комбинаторной модели “красно-синей игры с галькой”. В рамках проекта COSMA развивается проект Tiled-MM, представляющий собой наиболее близкий аналог алгоритма параллельного матричного

умножения, предложенного в данной диссертационной работе (однако Tiled-MM был опубликован заметно позже первой статьи с результатами данной диссертации).

Алгоритм матричного произведения применяется в различных вычислительных задачах, например, в алгоритме расчета матричной экспоненты. Задача расчета матричной экспоненты является вычислительно очень сложной задачей, поэтому для ее решения применяются различного рода упрощения, связанные с особыми характеристиками матриц. С другой стороны, вычислительные мощности суперкомпьютеров возрастают в наше время достаточно интенсивно. В результате уже не является чем-то совершенно невозможным рассмотрение задачи расчета матричной экспоненты в самом общем случае. Вариантов методов нахождения матричной экспоненты достаточно много. Их можно разделить на следующие категории: представление в виде суммы ряда, решение дифференциальных уравнений, полиномиальные методы, различного вида матричные разложения (в том числе и спектральное) или методы разделения [5; 6]. Как отмечают авторы процитированных обзоров, сложно сказать, который из представленных методов является “лучшим”. В случае для нормальных матриц автоматически за счет свойств нормальности пропадают многие проблемы, но в произвольном случае требуется бороться с ошибками округления.

При решении нестационарного уравнения Шредингера (TDSE) требуется многочисленный расчет матричной экспоненты, что делает эту задачу сложной для рассмотрения в общем случае. Решения частных задач TDSE можно найти в различных работах. Например, в работе Lugovskoy и Bray рассмотрено почти внезапное возмущение квантовой системы ультракоротким импульсом [7]: аналитическая теория сопоставлена с численным решением TDSE. Различные сценарии ионизации He были рассмотрены путем численного решения одномерного TDSE в работе Yu и Madsen [8; 9]. Неадиабатическая квантовая динамика молекулярных ионов H_2^+ и HD^+, возбуждаемых одиночными лазерными импульсами, линейно поляризованными вдоль оси молекулы, исследована в рамках трехмерной модели в работе Paramonov и др. [10]. Взаимодействие сильных лазерных полей с He, H_2^+ и H_2 было смоделировано на основе решения TDSE в работе Majorosi и др. [11]

Основные положения, выносимые на защиту:

1. Разработан алгоритм матричного умножения, использующий асинхронные обмены данными между несколькими графическими ускорителями в рамках одного сервера.
2. Построена теоретическая модель определения оптимального размера блока для предложенного алгоритма матричного умножения и выполнена ее проверка тестированием серверов с различной связью между графическими ускорителями.
3. Разработан алгоритм расчета матричной экспоненты, использующий несколько графических ускорителей, основанный на предложенном алгоритме матричного умножения.
4. Показано, что численное решение нестационарного одномерного уравнения Шредингера с использованием предложенного алгоритма расчета матричной экспоненты позволяет описывать неадиабатические переходы в моделях двухъядерных молекул с одним электроном.

Научная новизна: Разработан параллельный алгоритм матричного умножения для нескольких ГПУ, эффективность которого выше, чем у аналогов, входящих в стандартные библиотеки (cuBLASXt). Предложена аналитическая модель производительности данного алгоритма. Предсказательная сила модели проверена численными экспериментами на различных гибридных платформах. На основе разработанного алгоритма матричного умножения разработан алгоритм вычисления матричной экспоненты. С его помощью создана программа для решения зависящего от времени уравнения Шредингера и проведены расчеты молекулярного иона водорода. Показана возможность описания процесса неадиабатического перехода электронной подсистемы из основного в возбужденное состояние.

Методология и методы исследования.

Применялись инструменты программирования (Nvidia CUDA SDK, AMD ROCm), методы оптимизации, отладчики (Nsight, rocprof), связанные с программированием методы для составления программ и внешние математические библиотеки. Модель оптимального регулируемого параметра алгоритма (размера блоков) была получена аналитически. Разностные методы были применены для построения алгоритма решения зависящего от времени уравнения Шредингера.

Основные результаты исследования

Асинхронный алгоритм на нескольких ГПУ матричного произведения и матричной экспоненты

Разработана программа алгоритма матричного произведения общего вида

$$C = \alpha AB + \beta C, \tag{1}$$

использующая только графические ускорители для расчетов и хранения данных. В алгоритме обеспечена возможность хранить все данные в памяти одного устройства, обеспечивая остальные ГПУ данными для расчета, и хранения матриц в разных ускорителях, понижая нагрузку на процесс доступа к памяти.

Общая схема алгоритма следующая:

– устройства, хранящие матрицы A и B, отправляют полосы A_i и B_i другим графическим процессорам,
– ГПУ производят умножение матриц на α, используя полученные данные, и вычисляют блоки C'_{ij} , которые затем собираются в полосу C'_i,
– устройство, в котором хранится матрица C, собирает полосы C'_i и суммирует полученную матрицу с βC.

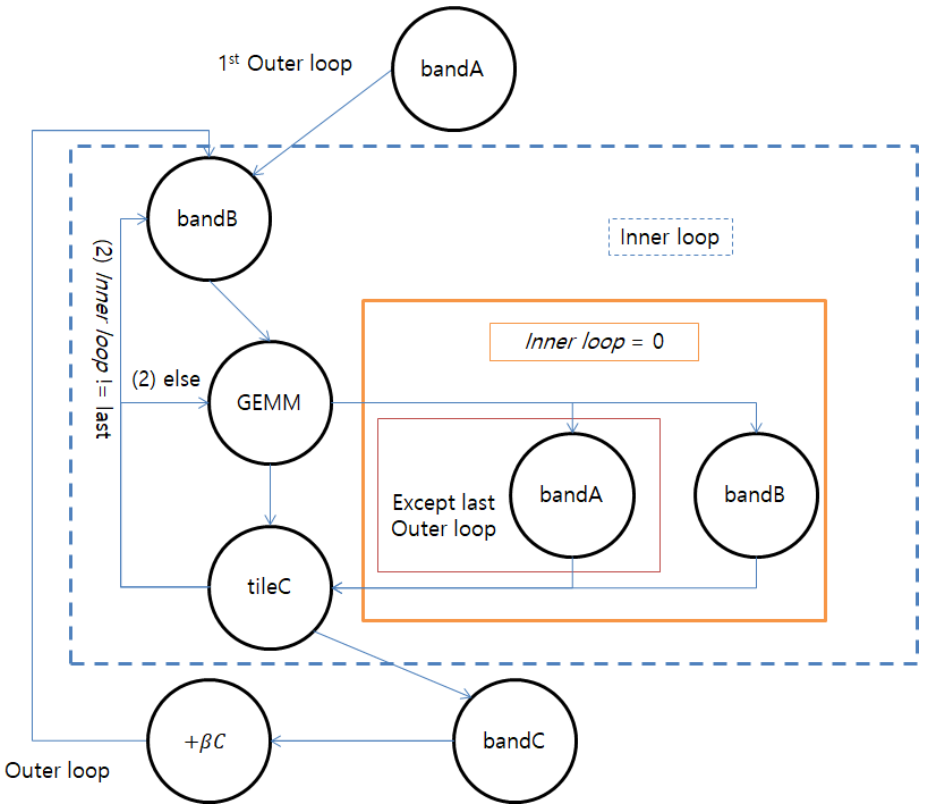

Рисунок 1 — Графическая схема алгоритма. Весь алгоритм работает в рамках внешнего цикла. Команды передачи данных могут вызываться в другом порядке, если проверено полное использование данных перед перезаписью.

Более детально схема работы алгоритма приводится в рис. 1.

Algorithm 1 Схематический процесс алгоритма Multi-GPU GEMM в произвольном рабочем ГПУ, m является количеством полос в строке (столбце).

```
for outerloop = 0 to m/NumOfGPUs do
  receive (bandA);
  for innerloop = 0 to m do
    recieve (bandB);
    GEMM (bandA, bandB, alpha, tileC);
    write_TileToBand (tileC, bandC);
  end for
  send (bandC);
end for
```

Алгоритм, выполняющийся на ускорителях, которые выполняют расчет, имеет следующий вид (см. Alg. 1):

Подробный алгоритм работы в ускорителях приведен в Alg. 2.

В общем случае экспонента матрицы A представляется как разложение ряда Тейлора

$$\exp A = \sum_{i=0}^{\infty} a_i = \sum_{i=0}^{\infty} \frac{1}{k!} A^k = I + \frac{1}{1!}A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \ldots, \tag{2}$$

где I — единичная матрица. Каждый k-ый элемент a_k выражается, как

$$a_k = \frac{1}{k} A a_{k-1}, \tag{3}$$

то есть для нахождения k-ого члена можно применить алгоритм (1) с матрицами A и a_{k-1} и константами $\alpha = 1/k, \beta = 0$.

В комплексном пространстве умножение двух комплексных матриц можно разбить на четыре отдельных умножения матриц с действительными числами

$$\begin{cases} Re(C) = \alpha(Re(A) * Re(B) - Im(A) * Im(B)) + \beta Re(C), \\ Im(C) = \alpha(Re(A) * Im(B) + Im(A) * Re(B)) + \beta Im(C). \end{cases} \tag{4}$$

В алгоритме матричной экспоненты $\beta = 0$ (3), поэтому на каждой итерации просто необходимо найти произведения действительных и мнимых частей с коэффициентом α и отдельно провести операции сложения или вычитания матриц. Для максимальной эксплуатации памяти по возможности каждые действительные и мнимые части матриц хранятся в отдельных графических ускорителях как отдельные матрицы.

Algorithm 2 Полный алгоритм Multi-GPU GEMM с учетом изменения команд и операции передачи данных для разных устройств.

```
if (device ≠ device_withA) then
  receive (bandA);
end if
for outerloop = 0 to m/NumOfGPUs do
  for innerloop = 0 to m do
    if (innerloop < m - 1) and (device ≠ device_withB) then
      recieve (bandB);
    end if
    GEMM (bandA, bandB, alpha, tileC);
    if (innerloop = 0) then
      if (device ≠ device_withB) then
        recieve (bandB);
      end if
      if (outerloop < m/NumOfGPUs - 1) and (device ≠ device_withA) then
        Receive (bandA);
      end if
    end if
  end for
  send (bandC);
  if (device == device_withC) then
    AddbetaC (C, beta, bandC);
  end if
end for
```

Теоретическая модель поиска оптимального размера блоков

Определена интенсивность расчета для случая квадратных матриц, при которых ограничения по скорости расчета будут происходить по вычислительной способности ускорителей, а не по пропускной способности памяти

$$\begin{cases} BW_{math}/BW_{mem} = k_{BW}, \\ Intensity = \frac{N_i^2 N}{2(N_i N + N_i N + N_i^2)} = \frac{N_i N}{4N + 2N_i} > k_{BW}. \end{cases} \tag{5}$$

где BW — вычислительная способность процессора (math) или пропускная способность памяти (mem), соответственно. Отсюда получено условие на размер

блоков

$$N_i > 4k_{BW}N/(N - 2k_{BW}),\ T_{math} > T_{mem}. \tag{6}$$

В работе рассматриваются достаточно большие размеры матриц, требующих больших вычислительных мощностей. В этом случае получено время передачи данных между устройствами

$$T_{transfer} = 4N_iN/BW_{transfer}. \tag{7}$$

Если задача упирается на вычислительные мощности ускорителей, при условии на размеры блоков

$$N_i > 2(Num_{GPUs} - 1)BW_{math}/BW_{transfer}, \tag{8}$$

скорость расчета должен быть больше, чем скорость поставки данных, и не должно происходить ожидание. Здесь Num_{GPUs} — количество графических ускорителей, применяемых в расчете. Время расчета линейно зависит от размера блока, поэтому для оптимального расчета накладываются условия

$$\begin{cases} N_i > 4k_{BW}N/(N - 2k_{BW}), & N > 2k_{BW}, \\ N_i > 2(Num_{GPUs} - 1)BW_{math}/BW_{transfer}, \\ N_i \to \min. \end{cases} \tag{9}$$

Разработанная модель оптимального размера блоков была проанализирована и проверена профилировщиком. Прогнозированные размеры блоков были подтверждены для алгоритма произведения квадратных матриц, равномерно распределяющего задачи на ускорители.

Вычислительные эксперименты на разных аппаратных обеспечениях

Результаты были собраны на платформах с 4 V100, связанных NVLink 2.0; 8 A100, связанных NVLink 3.0; 4 GTX 1070, связанных PCIe 3.0 и 4 RX 6900 XT, связанных PCIe 4.0.

На платформах с четырьмя V100 и с четырьмя GTX1070 были показаны удовлетворение модели оптимального параметра.

В случае работы с графическими ускорителями потребительского уровня (AMD RX 6900 XT) были замечены потери производительности и задержки между передачами данных (см рис. 2).

На примере с значимо влияющими факторами, как неравномерная балансировка вычислительной нагрузки на каждые ГПУ, которые могут возникнуть

в зависимости от рассматриваемых параметров задачи, продемонстрировано для сервера на базе восьми A100 с NVLink возможное отклонение эмпирически оптимальных параметров от предсказанных. На примере этой же платформы показано, как использование тензорных ядер меняет баланс между связью и вычислениями (см. рис. 3).

Использование разработанного алгоритма для решения одномерного зависящего от времени уравнения Шредингера

Разработан алгоритм решения одномерного нестационарного уравнения Шредингера следующим образом (Alg. 3).

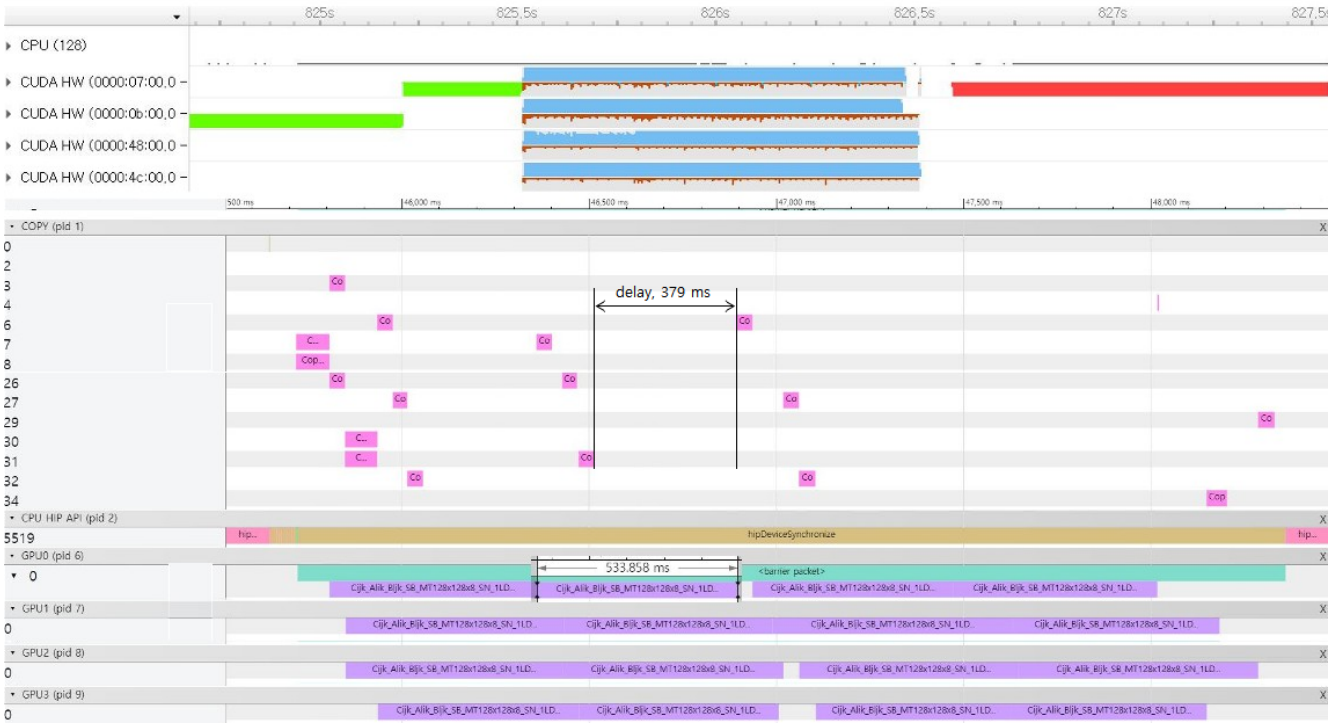

Рисунок 2 — Профили работ Multi-GPU GEMM на 4 A100 ГПУ без тензорных ядер (вверху) и на 4 RX 6900 XT ГПУ (внизу). Количество элементов ($N = 32768$) в строке (столбце) матриц и размер блоков ($N_i = 1024$) в случае A100 ГПУ и ($N_i = 8192$) в случае RX 6900 XT ГПУ. Матрицы A, B, и C хранятся в устройствах 2, 1, 0 соответственно. Зеленые столбцы (вверху, внизу пропущено) — это передача данных от хоста к ускорителям для расчетов, красные столбцы (вверху, внизу пропущено) — это передача результирующих данных от ускорителей к хосту, синие столбцы (вверху) и фиолетовые столбцы (внизу) — это вычисления в ГПУ и коричневые столбцы (вверху) и розовые столбцы (внизу) — это операции передачи данных между ускорителями типа peer-to-peer.

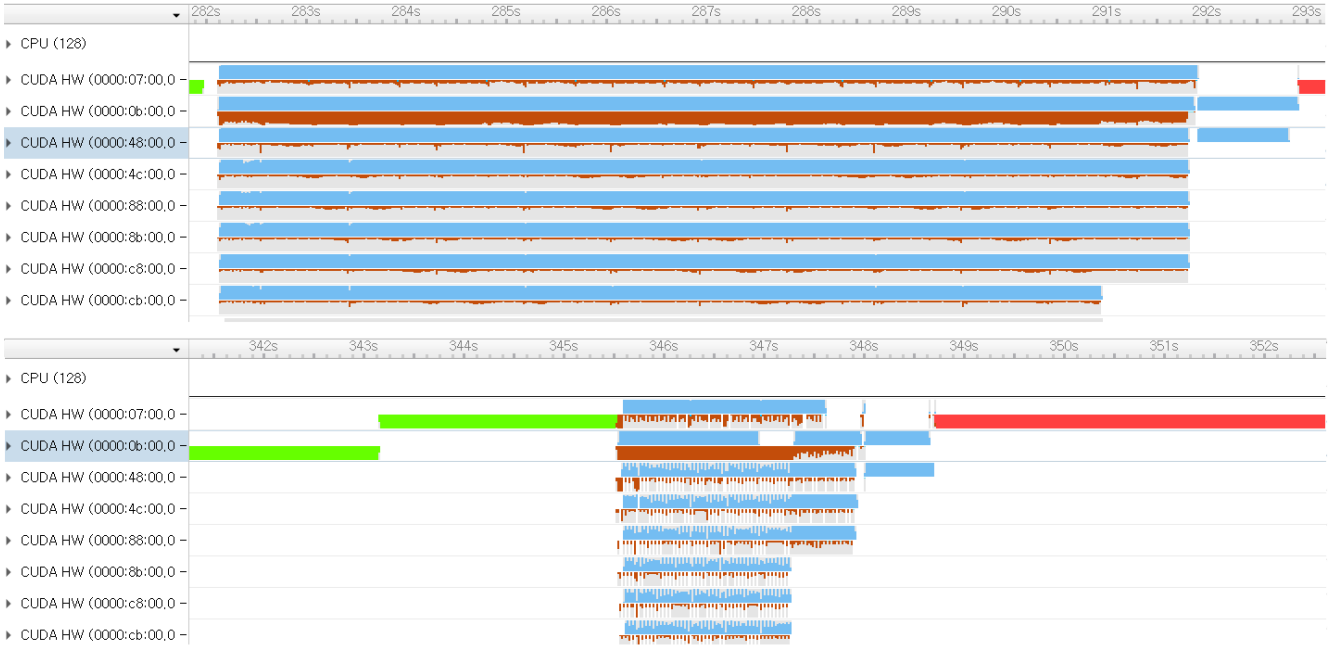

Рисунок 3 — Профили работ Multi-GPU GEMM на 8 A100 ГПУ без тензорных ядер (вверху) и с тензорными ядрами (внизу). Количество элементов ($N = 90000$) в строке (столбце) матриц и размер блоков ($N_i = 1024$) в случае без тензорных ядер и ($N_i = 4096$) в случае с тензорными ядрами. Матрицы A, B, и C хранятся в устройствах 2, 1, 0 соответственно. Зеленые столбцы — это передача данных от хоста к ускорителям для расчетов, красные столбцы — это передача результирующих данных от ускорителей к хосту, синие столбцы — это вычисления в ГПУ и коричневые столбцы — это операции передачи данных между ускорителями типа peer-to-peer.

Найдено решение стационарного случая модели молекулярного иона водорода (два протона и один электрон) со смягченным кулоновским потенциалом (soft-core Coulomb potential, см. рис. 4).

$$V(r) = \frac{-Z}{\sqrt{r^2 + a}}. \tag{10}$$

Для имитации ионной вибрации мы реализовали движение ионных центров.

$$V(r,t) = \frac{-Z}{\sqrt{(r - \alpha \sin(\beta t))^2 + a}}, \tag{11}$$

где Z — сила кулоновского взаимодействия мягкого ядра, a — параметр смягчения [8], α и β — некоторые константы.

Установлен эффект при расчете комплексной матричной экспоненты (рис. 5), из-за которого происходит рост погрешности. Для устранения этой проблемы необходимо соблюдать соотношение шагов по времени и расстоянию $\Delta r \sim \Delta t^2$.

Algorithm 3 Схема алгоритма решения нестационарного уравнения Шредингера, выполняемая на графических процессорах за одну временную итерацию, ψ — волновая функция, H — сгенерированная матрица Гамильтона, $\exp A$ — полученная матричная экспонента. Все операции выполняются отдельно для вещественных и мнимых матриц. Сообщение между графическими ускорителями не показано.

receive (potential V);
build Hamiltonian matrix ($H = T + V$), $A = iH\Delta t$;
$\exp A = I + A$;
for $k = 2$ to rank **do**
  $\alpha = 1/k$;
  Multi-GPU GEMM ($A^k = \alpha A A^{k-1}$);
  $\exp A = \exp A + A^k$;
end for
GEMV ($\psi_j = \exp A \; \psi_{j-1}$);
send (ψ_j); //for output in file

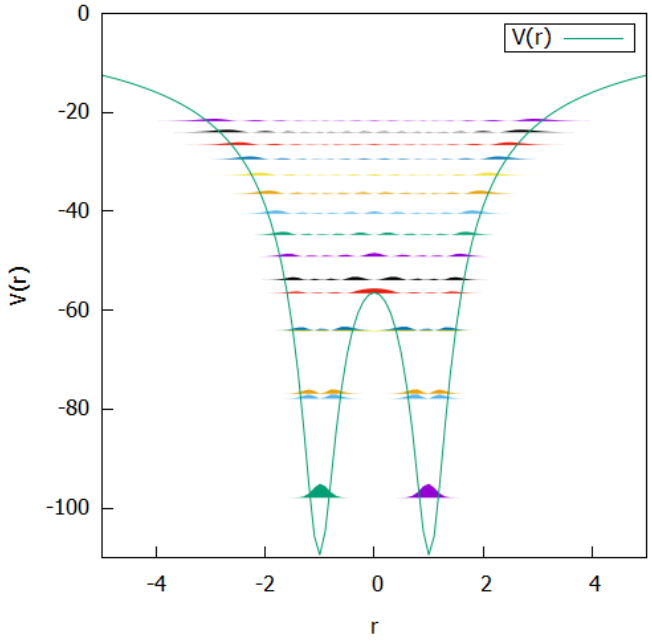

Рисунок 4 — Пример распределения электронной плотности $|\psi_n(r)|^2$ для нескольких связанных состояний на соответствующих положениях энергетических уровней в кулоновском потенциале мягкого ядра двух ионов (10) ($Z = 30, a = 0.1$).

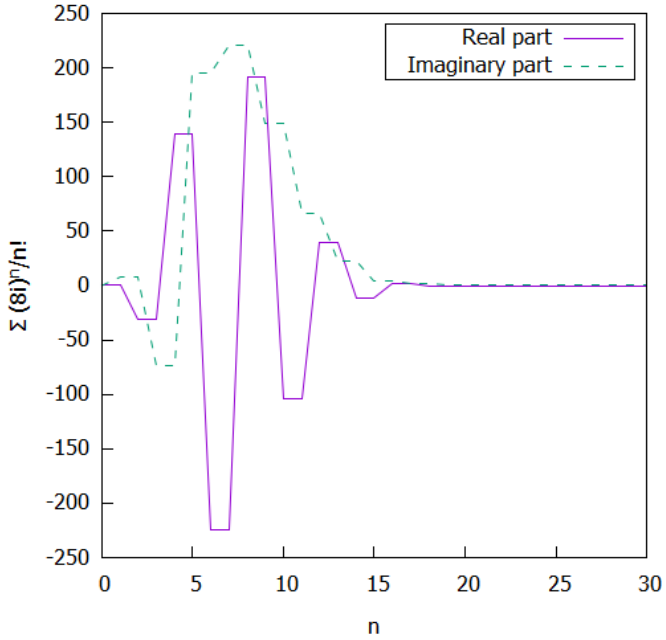

Рисунок 5 — Поведение функции $\sum\limits_{n} \frac{(8i)^n}{n!}$, иллюстрирующее типичную сходимость элементов матричного показателя степени.

Получены результаты экспериментов неподвижными и движущимися синусоидально ионами (центрами потенциала) молекулярного иона водорода. В рассмотренном нестационарном случае был наблюден неадиабатический переход электронной подсистемы из основного в возбужденное состояние (рис. 6).

Выводы

Основные результаты работы заключаются в следующем.

1. Разработан асинхронный алгоритм Multi-GPU GEMM умножения матриц на нескольких графических ускорителей с коммуникацией только между ускорителями без обращения в память центрального процессора.
2. Разработан алгоритм матричной экспоненты версии с действительными и с комплексными числами на основе алгоритма Multi-GPU GEMM.
3. Выведена теоретическая модель, предсказывающая оптимальный варьируемый параметр размера блоков, при которых алгоритм Multi-GPU GEMM показывает наилучшую производительность. Установлено, что отпимальный размер зависит от платформозависимых параметров.
4. Реализована HIP версия алгоритма Multi-GPU GEMM для работы с графическими ускорителями AMD.

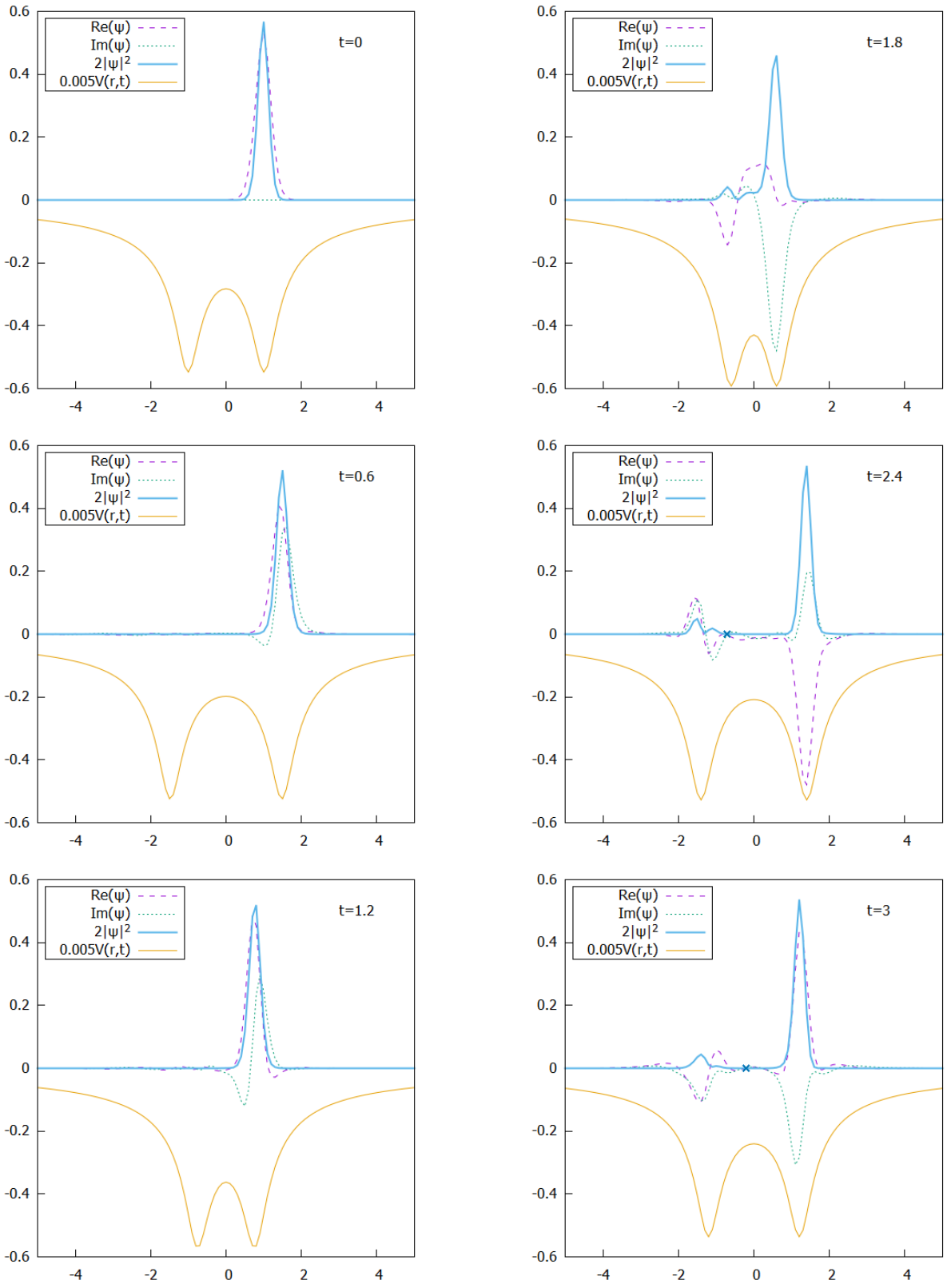

Рисунок 6 — Поведение комплексной волновой функции (ψ) в синусоидально движущемся двухямном кулоновском потенциале с мягким ядром (11) (начальное расстояние между ямами равно 2, $Z = 30, a = 0.1$, $\Delta t = 0{,}002$, $\Delta r = 0{,}1$, $k = 10$, $\alpha = 0{,}5$ и $\beta = 3$). Точками обозначены узлы волновой функции, возникающие после возбуждения из основного состояния.

5. Алгоритм Multi-GPU GEMM был успешно запущен и протестирован на разных вычислительных платформах. Достигнута высокая производительность свыше 80% от пиковой для ускорителей серверного и примерно 40% для потребительского уровня.
6. Проверены прогнозируемые значения варьируемых параметров предложенной модели по эмпирическим результатам. Описаны факторы, влияющие на возможные расхождения для реальных случаев работы с алгоритмом.
7. Разработан параллельный алгоритм на нескольких графических ускорителей решения одномерного зависящего от времени уравнения Шредингера на базе вышеописанных алгоритмов.
8. Промоделирован неадиабатический перенос энергии от движущихся ядер к одноэлектронному возбуждению иона водорода H_2^+.

Апробация работы.

Результаты работы были доложены на следующих конференциях.

1. Цифровая индустрия: состояние и перспективы развития 2020 (ЦИСП'2020), Россия, Челябинск, 17-19 ноября, 2020. Доклад «Matrix-Matrix Multiplication Using Multiple GPUs Connected by Nvlink».
2. 20-я международная конференция и молодежная школа Математическое моделирование и суперкомпьютерные технологии (ММиСТ2020) в рамках Международных конгресса «Суперкомпьютерные дни в России», Нижегородский Государственный Университет им. Н.И. Лобачевского, Россия, Нижний Новгород, 23-25 ноября, 2020. Доклад «Алгоритм матричного умножения для нескольких GPU, объединенных высокоскоростными каналами связи».
3. 9-я международная конференция "Distributed Computing and Grid Technologies in Science and Education"(GRID'2021), Россия, Дубна, 5-9 июля, 2021. Доклад «Overlapping Computation and Communication in Matrix-Matrix Multiplication Algorithm for Multiple GPUs».
4. Международная конференция Параллельные вычислительные технологии (ПаВТ) 2022, Россия, Дубна, 29-31 марта, 2022. Доклад «The Tuning of Matrix-Matrix Multiplication Algorithm for Several GPUs Connected by Fast Communication Links».
5. Летняя школа «Extreme-scale big data analytics and scientific computing on heterogeneous platforms», Lake Como School of Advanced Studies,

Como, Italy, 26-30 сентября, 2022. Доклад «Multi-GPU GEMM algorithm: maximizing efficiency on different platforms».

6. 22-я международная конференция и молодежная школа Математическое моделирование и суперкомпьютерные технологии (ММиСТ 2022), Нижегородский Государственный Университет им. Н.И. Лобачевского, Нижний Новгород, 14-17 ноября, 2022. Доклад «Multi-GPU GEMM algorithm performance analysis for Nvidia and AMD GPUs connected by NVLink and PCIe».
7. The Conference on Computational Physics (CCP2023) – 34th International Union of Pure and Applied Physics (IUPAP) Conference on Computational Physics, Kobe International Conference Center, Kobe, Japan, 4-8 августа, 2023. Доклад «Multi-GPU GEMM for 1D Time-Dependent Schrodinger Equation».
8. Международная конференция «Суперкомпьютерные дни в России 2023», Россия, Москва, 25-26 сентября, 2023. Доклад «GPU-accelerated matrix exponent for solving 1D time-dependent Schrodinger equation».

Личный вклад.

Под взглядом научного руководителя на общую картину и структуру работы соискатель составил все приводимые алгоритмы и написал соответствующие программы. Соискатель провел вычислительные эксперименты на разных платформах, доступ к которым был обеспечен научным руководителем. С помощью руководителя были проанализированы результаты численных экспериментов, и совместно разработаны идеи для устранения уязвимостей и модификации программы в лучшую сторону, которые затем внедрялись в код соискателем. Соискатель лично делал все доклады на конференциях разных форматов. Соискателем обеспечивались данные, приводимых в публикациях, и были написаны основная часть первой статьи, и большая часть, включая основную, остальных статей, которые затем подвергались осмотру и корректировке научным руководителем.

Публикации. Основные результаты по теме диссертации изложены в 4 печатных изданиях, 4 — в периодических научных журналах, индексируемых Web of Science и Scopus.

Публикации автора по теме диссертации

1. *Choi, Y. R.* Matrix-Matrix Multiplication Using Multiple GPUs Connected by NVLink [Текст] / Y. R. Choi, V. Nikolskiy, V. Stegailov // 2020 Global Smart Industry Conference (GloSIC). — IEEE. 2020. — С. 354—361.
2. *Choi, Y. R.* Multi-GPU GEMM Algorithm Performance Analysis for Nvidia and AMD GPUs Connected by NVLink and PCIe [Текст] / Y. R. Choi, V. Stegailov // Mathematical Modeling and Supercomputer Technologies: 22nd International Conference, MMST 2022, Nizhny Novgorod, Russia, November 14–17, 2022, Revised Selected Papers. — Springer. 2022. — С. 281—292.
3. *Choi, Y. R.* Tuning of a Matrix-Matrix Multiplication Algorithm for Several GPUs Connected by Fast Communication Links [Текст] / Y. R. Choi, V. Nikolskiy, V. Stegailov // International Conference on Parallel Computational Technologies. — Springer. 2022. — С. 158—171.
4. *Choi, Y. R.* GPU-Accelerated Matrix Exponent for Solving 1D Time-Dependent Schrödinger Equation [Текст] / Y. R. Choi, V. Stegailov // Supercomputing / под ред. V. Voevodin [и др.]. — Cham : Springer Nature Switzerland, 2023. — С. 100—113.

Список литературы

1. *Schulthess, T. C.* Programming revisited [Текст] / T. C. Schulthess // Nature Physics. — 2015. — Т. 11, № 5. — С. 369—373.
2. BLASX: A high performance level-3 BLAS library for heterogeneous multi-GPU computing [Текст] / L. Wang [и др.] // Proceedings of the 2016 International Conference on Supercomputing. — 2016. — С. 1—11.
3. Generic matrix multiplication for multi-GPU accelerated distributed-memory platforms over PaRSEC [Текст] / T. Herault [и др.] // 2019 IEEE/ACM 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA). — IEEE. 2019. — С. 33—41.

4. Red-blue pebbling revisited: near optimal parallel matrix-matrix multiplication [Текст] / G. Kwasniewski [и др.] // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. — 2019. — С. 1—22.

5. *Moler, C.* Nineteen dubious ways to compute the exponential of a matrix [Текст] / C. Moler, C. Van Loan // SIAM review. — 1978. — Т. 20, № 4. — С. 801—836.

6. *Moler, C.* Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later [Текст] / C. Moler, C. Van Loan // SIAM review. — 2003. — Т. 45, № 1. — С. 3—49.

7. *Lugovskoy, A.* Almost sudden perturbation of a quantum system with ultrashort electric pulses [Текст] / A. Lugovskoy, I. Bray // Physical Review A. — 2008. — Т. 77, № 2. — С. 023420.

8. *Yu, C.* Sequential and nonsequential double ionization of helium by intense XUV laser pulses: Revealing ac Stark shifts from joint energy spectra [Текст] / C. Yu, L. B. Madsen // Physical Review A. — 2016. — Т. 94, № 5. — С. 053424.

9. *Yu, C.* Above-threshold ionization of helium in the long-wavelength regime: Examining the single-active-electron approximation and the two-electron strong-field approximation [Текст] / C. Yu, L. B. Madsen // Physical Review A. — 2017. — Т. 95, № 6. — С. 063407.

10. Quantum dynamics, isotope effects, and power spectra of H_2^+ and HD^+ excited to the continuum by strong one-cycle laser pulses: Three-dimensional non-Born-Oppenheimer simulations [Текст] / G. K. Paramonov [и др.] // Physical Review A. — 2018. — Т. 98, № 6. — С. 063431.

11. Density-based one-dimensional model potentials for strong-field simulations in He, H_2^+, and H_2 [Текст] / S. Majorosi [и др.] // Physical Review A. — 2020. — Т. 101, № 2. — С. 023405.