

В.А. Лохвицкий, А.И. Захаров, Г.А. Брякалов, К.А. Неретина

ИССЛЕДОВАНИЕ УСКОРЕНИЯ ОБРАБОТКИ ДАННЫХ В МНОГОЯДЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ НА ОСНОВЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Аннотация. Статья посвящена разработке имитационной модели гипотетической вычислительной задачи, а также исследованию на ее основе возможностей различных программно-аппаратных решений с точки зрения затрачиваемых на организацию вычислений временных ресурсов. Необходимость такой работы обуславливается требованиями, предъявляемыми к скорости обработки больших объемов информации, появлением новых аппаратно-программных средств распараллеливания и потребностью в их апробации. Особое значение в организации исследования имеет метод имитационного моделирования, применение которого обусловлено малоизученностью рассматриваемых аппаратно-программных платформ, а также необходимостью предварительных оценок их дальнейшего использования при решении трудоемких вычислительных задач.

Ключевые слова: имитационное моделирование, распараллеливание алгоритмов, параллельные вычисления, графические процессоры.

V.A. Lokhvitsky, A.I. Zakharov, G.A. Bryakalov, K.A. Neretina

DATA PROCESSING ACCELERATION STUDY IN MULTI-CORE COMPUTING BASED ON SIMULATION

Abstract. The article is devoted to the development of a simulation model of a hypothetical computational problem, as well as to the study on its basis of the possibilities of various software and hardware solutions in terms of time resources spent on organizing calculations. The need for such work is caused by the requirements imposed on the processing speed of large amounts of information, the emergence of new hardware and software parallelization tools and the need for their approbation. Of particular importance in the organization of the study is the method of simulation modeling, the use of which is due to the little-studied hardware and software platforms under consideration, as well as the need for preliminary assessments of their further use in solving time-consuming computational tasks.

Keywords: simulation modeling, parallelization of algorithms, parallel computing, graphics processors.

Введение

Мировая практика принятия сложных управленческих решений перешла на принципиально новый уровень методологической и инструментальной поддержки, когда те или иные варианты решений должны быть предварительно апробированы не на реальных объектах и людях, а на их аналогах, то есть на моделях [10]. В этой связи осуществление экономических, технических решений или новаций требует предварительных оценок финишных результатов при помощи имитационного моделирования (далее – ИМ) [10]. Этим объясняется актуальность темы данной работы.

Имитационное моделирование – это метод исследования, при котором изучаемая система заменяется моделью, достаточно точно описывающий реальную систему, с которой проводятся эксперименты для получения информации об этой системе [10].

Лохвицкий Владимир Александрович

кандидат технических наук, заместитель начальника кафедры математического и программного обеспечения. Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург. Область научных интересов: информационные технологии; информатика; организация вычислительного процесса; программирование; системный анализ. Автор 90 опубликованных научных работ. Электронный адрес: vka_kaf27_1@mil.ru

Захаров Анатолий Иванович

кандидат технических наук, профессор, доцент кафедры математического и программного обеспечения. Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург. Область научных интересов: информационные технологии; организация параллельного вычислительного процесса; параллельное программирование. Автор более 100 опубликованных научных работ. Электронный адрес: vka_kaf27_1@mil.ru

Брякалов Геннадий Алексеевич

кандидат технических наук, доцент, доцент кафедры математического и программного обеспечения. Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург. Область научных интересов: информационные технологии; информатика; организация вычислительного процесса; программирование. Автор более 100 опубликованных научных работ. Электронный адрес: vka_kaf27_1@mil.ru

Неретина Кристина Андреевна

инженер учебной лаборатории. Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург. Область научных интересов: математика; информационные технологии; параллельное программирование. Автор 5 опубликованных научных работ. Электронный адрес: vka_kaf27_1@mil.ru

Работа посвящена разработке имитационной модели, позволяющей исследовать процесс ускорения обработки данных, а также влияние аппаратных средств распараллеливания алгоритма гипотетической задачи на повышение оперативности обработки информации в многоядерных компьютерах.

Разработка алгоритма модели

При исследовании предельных возможностей различных программно-аппаратных решений, используемых в процессе организации вычислительно-трудоемких задач, возникает необходимость проведения ряда экспериментов. Так, при имитационном моделировании алгоритм, реализующий конкретную модель, воспроизводит процесс функционирования системы во времени и пространстве. В этом случае имитируются элементарные явления с сохранением логической и временной структуры процесса [5]. С этой точки зрения задачу синтеза целесообразно решать методом имитационного моделирования.

При разработке имитационной модели типовой процесс моделирования содержит следующие основные этапы [10].

1. Создание концептуальной модели.
2. Разработка алгоритма модели.
3. Программирование модели.
4. Планирование прогонов.

5. Машинный эксперимент.

6. Анализ результатов.

Концептуальная модель включает в себя:

- целевые показатели;
- описание подсистем и их взаимодействия;
- числовые данные.

В целом в качестве объекта имитационной модели вычислительной задачи может использоваться гипотетическая задача, схема алгоритма которой, представлена на Рисунке 1.

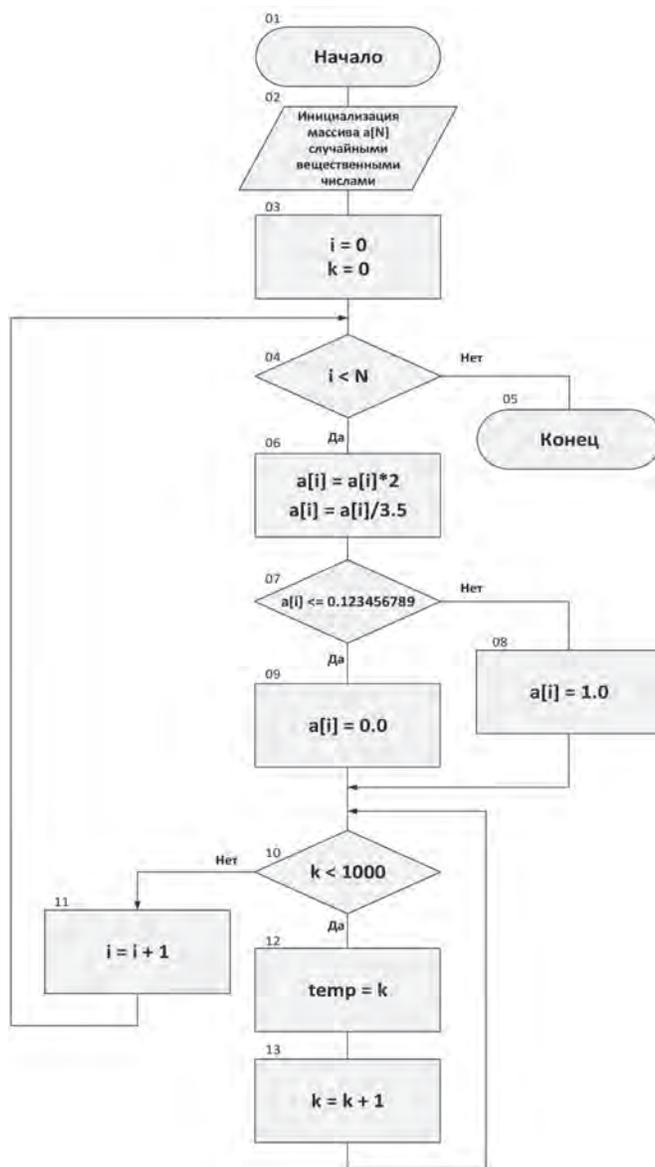


Рисунок 1. Схема алгоритма имитационной модели

Содержательная постановка задачи

Содержательная постановка задачи включает в себя описание аппаратной части исследования, алгоритм обработки данных и языки программирования.

Аппаратная часть базируется на современной технологии CUDA (англ. Compute Unified Device Architecture), получившей широкое распространение. Эта программно-аппаратная архитектура позволяет производить вычисления с использованием графических процессоров NVIDIA, поддерживающих технологию GPGPU [1].

В основе преимуществ использования технологии CUDA лежат архитектурные особенности построения графических процессоров (GPU), отличающие их от традиционных центральных процессоров (CPU) [11].

Выполнение расчетов на GPU показывает отличные результаты в алгоритмах, использующих параллельную обработку данных, особенно когда одну и ту же последовательность математических операций применяют к большому объему независимых блоков данных. При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико [11].

Основой для эффективного использования возможностей GPU в научных и иных неграфических расчетах является распараллеливание алгоритмов на сотни исполнительных блоков, имеющих в графических процессорах. Что касается технологии CUDA, она базируется на расширении с помощью видеокарт NVIDIA языков C++ и Python для параллельных вычислений. Эффект от применения видеокарт выражается в простоте скорости вычислений по сравнению с центральными процессорами общедоступных компьютеров [1].

Одной из основных проблем параллельных вычислений является определение возможности распараллеливания существующих последовательных алгоритмов решения задачи, а также непосредственно само распараллеливание алгоритмов с их дальнейшей программной реализацией [2–4].

Суть программного распараллеливания составляет декомпозиция – разбиение задачи на отдельные независимые части и параллельная их реализация на многоядерной вычислительной системе (далее – МВС).

Алгоритм имитационной модели был реализован в трех вариантах программно-аппаратных конфигураций вычислительной системы и отражает пошаговый процесс создания имитационной модели генерирования некоторой гипотетической задачи [5; 7; 9].

В первом варианте использовалась библиотека Numba языка Python, позволяющая применять графический адаптер с технологией CUDA для ускорения процесса вычислений.

Во втором варианте алгоритм был реализован на языке C++, а вычисления выполнялись в однопоточном режиме с использованием центрального процессора (CPU).

Третий вариант – это реализация алгоритма на языке Python версии 3.8 с использованием библиотеки Numpy и выполнением программы с использованием центрального процессора.

Конфигурация используемого в эксперименте ноутбука включала в себя:

- процессор Intel Core i7 8th Gen, ОЗУ – 16 Гб;
- видеоадаптер NVIDIA GeForce MX150;
- операционная система MS Windows 10 Домашняя x64.

Математическая постановка задачи

Дано:

1. Программа имитационной модели в кодах языка программирования Python версии 3.8.
2. Конфигурация ноутбука, указанная выше.

Требуется:

1. Реализовать программу в трех вариантах программно-аппаратных конфигураций вычислительной системы:

- GPU (CUDA), Python3, Numpy, Numba;
- CPU, C++;
- CPU, Python3, Numpy.

2. Провести сравнительный анализ времени решения трех вариантов проведенного эксперимента по ускорению обработки данных [6; 8].

В ходе проведения эксперимента результаты сравнительного анализа времени решения задачи были сведены в Таблицу 1, а также представлены на графике Рисунка 2.

Таблица 1

Времена вычислений в зависимости от объема входных данных (N)

Размер массива, N	Время, с		
	GPU, Numba	CPU, C++	CPU, Python3
100	0,0006	0,0002	0,0070
1000	0,0016	0,0028	0,0690
10000	0,0009	0,0213	0,6725
100000	0,0021	0,2004	8,6524
1000000	0,0127	1,6913	75,0906
10000000	0,1159	17,1848	731,8543

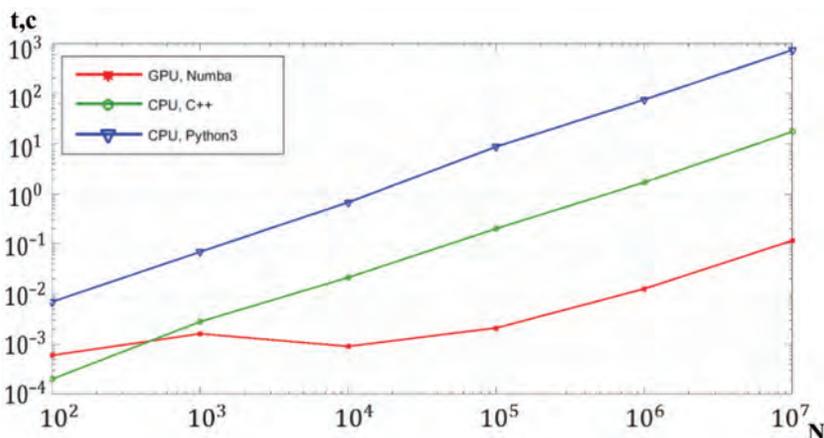


Рисунок 2. Зависимость длительности обработки данных от их объема при различных программно-аппаратных вариантах организации вычислений

В качестве входных данных использовался массив вещественных случайных чисел из диапазона [0,0; 1,0] размерности $N = \{10^2, 10^3, \dots, 10^7\}$.

В результате эксперимента установлено следующее:

- использование первого варианта конфигурации (GPU, Numba) позволяет обеспечить достаточно высокую оперативность решения задачи на всем диапазоне объемов входных данных;
- реализация алгоритма на C++ сопоставима по скорости обработки с первым вариантом только при малых объемах входных данных, а при их увеличении проигрывает на несколько порядков;
- использование «чистого» языка Python3 для реализации алгоритма и вычислений на центральном процессоре представляется нецелесообразным (для $N = 10^7$ данный вариант медленнее первого варианта в 6314,53 раз).

Заключение

Таким образом, в результате проведенного экспериментального исследования, можно сделать следующие выводы:

- снижение длительности решения задачи возможно на основе использования вычислительных возможностей графического ускорителя применительно к частям программы, допускающих параллельное выполнение;
- реализации вычислительно-трудоемких задач могут быть выполнены на языке Python с использованием библиотеки Numba;
- вычислительно-трудоемкие операции (допускающие независимую параллельную обработку) могут быть объединены в рамках отдельного вычислительного модуля (веб-сервиса), доступ к которому целесообразно реализовать на основе унифицированного интерфейса, например, по протоколу *jsonrpc*.

Статья носит прикладной характер и может быть полезна для лиц, интересующихся вопросами повышения оперативности обработки информации.

Литература

1. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA. М.: ДМК-Пресс, 2016. 232 с.
2. Воеводин В.В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 584 с.
3. Гергель В.П. Теория и практика параллельных вычислений. М.: Интернет-Университет, БИНОМ. Лаборатория знаний, 2007. 408 с.
4. Горелик А.М. Средства поддержки параллельности в языках программирования // Открытые системы. 1995. № 2. С. 26.
5. Захаров А.И., Пореченский М.А., Чмыхова Я.В. Имитационная модель исследования влияния распараллеливания информационных процессов на рост производительности многоядерных вычислительных систем // Сборник алгоритмов и программ прикладных задач. 2017. Вып. 34. С. 173–181.
6. Захаров А.И., Лохвицкий В.А., Старобинец Д.Ю., Хомоненко А.Д. Оценка влияния параллельной обработки изображений на оперативность функционирования БКУ КА дистанционного зондирования Земли // Современные проблемы дистанционного зондирования Земли из космоса. 2019. Т. 16, № 1. С. 61–71.
7. Захаров А.И., Брякалов Г.А., Неретина К.А. Влияние параллельных вычислений и структуры алгоритмов решаемых задач на оперативность обработки информации в многопроцессорных вычислительных системах // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2021. Вып. 1. С. 143–149.

8. Лохвицкий В.А., Борозенец А.Г. Программный комплекс классификации неструктурированных данных на основе метода опорных векторов // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2015. Вып. 1. С. 27–31.
9. Лохвицкий В.А., Захаров Р.С. Модель оценивания оперативности многопоточной обработки задач в распределенной вычислительной среде с учетом процессов Split-Join // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2019. Вып. 1. С. 26–34.
10. Рыжиков Ю.И. Имитационное моделирование. Курс лекций. СПб.: ВКА имени А.Ф. Можайского, 2007. 125 с.
11. Сандерс Д., Кэндрот Э. Технология CUDA в примерах. Введение в программирование графических процессов. М.: ДМК-Пресс, 2015. 232 с.

References

1. Boreskov A.V., Kharlamov A.A. (2016) *Osnovy raboty s tekhnologiyey CUDA* [Fundamentals of working with the technology of KUDA]. Moscow, DMK-Press Publishing, 232 p. (in Russian).
2. Voevodin V.V., Voevodin V.I. (2002) *Parallelnye vychisleniya* [Parallel computing]. St. Petersburg, BKhV-Peterburg Publishing, 584 p. (in Russian).
3. Gergel' V.P. (2007) *Teoriya i praktika parallelnykh vychislenij* [Theory and practice of parallel computing]. Moscow, BINOM, Laboratoriya znaniy, 464 p. (in Russian).
4. Gorelik A.M. (1995) *Sredstva podderzhki parallel'nosti v yazykakh programmirovaniya* [Parallelism Support Tools in Programming Languages]. *Otkrytye sistemy*, No. 2, p. 26 (in Russian).
5. Zakharov A.I., Porechenskiy M.A., Chmykhova Ya.V. (2017) *Imitatsionnaya model' issledovaniya vliyaniya rasparallelivaniya informatsionnykh protsessov na rost proizvoditel'nosti mnogoyadernykh vychislitel'nykh system* [Imitatsionnaya model' issledovaniya vliyaniya rasparallelivaniya informatsionnykh protsessov na rost proizvoditel'nosti mnogoyadernykh vychislitel'nykh system]. *Sbornik algoritmov i program prikladnykh zadach*, vol. 34, pp. 173–181 (in Russian).
6. Zakharov A.I., Lokhvitskiy V.A., Starobinets D.Yu., Khomonenko A.D. (2019) *Otsenka vliyaniya parallelnoy obrabotki izobrazheniy na operativnost' funktsionirovaniya BKU KA distantsionnogo zondirovaniya Zemli* [Estimation of the influence of parallel image processing on the efficiency of the functioning of the OCU of the Earth remote sensing spacecraft]. *Sovremennye problem distantsionnogo zondirovaniya Zemli iz kosmosa*, vol. 16, No. 1, pp. 61–71 (in Russian).
7. Zakharov A.I., Bryakalov G.A., Neretina K.A. (2021) *Vliyanie paralelnykh vychislenij i struktury algoritmov reshaemykh zadach na operativnost' obrabotki informatsii v mnogoprotsessornykh vychislitel'nykh sistemakh* [Influence of Parallel Computations and the Structure of the Algorithms of Solved Problems on the Efficiency of Information Processing in Multiprocessor Computing Systems] *Vestnik Rossiyskogo novogo universiteta. Seriya: Slozhnye sistemy: modeli, analiz i upravlenie*, vol. 1, pp. 143–149 (in Russian).
8. Lokhvitskiy V.A., Borozenets A.G. (2015) *Programmnyi kompleks klassifikatsii nestructurirovannykh dannykh na osnove metoda opornykh vektorov* [software package for classification of unstructured data based on the support vector machine]. *Vestnik Rossiyskogo novogo universiteta. Seriya: Slozhnye sistemy: modeli, analiz i upravlenie*, vol. 1, pp. 27–31 (in Russian).
9. Lokhvitskiy V.A., Khabarov R.S. (2019) *Model' otsenivaniya operativnosti mnogopotchnoy obrabotki zadach v raspredelennoi vychislitel'noi srede s uchetom protsessov Split-Join* [A model for evaluating the efficiency of multi-threaded processing of tasks in a distributed computing environment, taking into account Split-Ein processes]. *Vestnik Rossiyskogo novogo universiteta. Seriya: Slozhnye sistemy: modeli, analiz i upravlenie*, vol. 1, pp. 26–34 (in Russian).

10. Ryzhikov Yu.I. (2007) *Imitatsionnoe modelirovanie. Kurs leksij* [Simulation modeling. Lecture course]. St. Petersburg, VKA imeni A.F. Mozhayskogo, 125 p. (in Russian).
11. Sanders D., Kendrot E. (2015) *Tekhnologiya CUDA v primerakh. Vvedenie v programmirovaniye graficheskikh protsessov* [MIRACLE technology in examples. Introduction to Graphics Programming]. Moscow, DMK-Press, 232 p. (in Russian).