

Поиск оптимального маршрута движения горного транспорта с помощью технологий параллельных вычислений

А. Н. Ильюшина

Санкт-Петербургский колледж управления
и коммерции
bdbyu@mail.ru

И. М. Новожилов

Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
novozhilovim@list.ru

Аннотация. В статье проведен сравнительный анализ современных технологии организации параллельных вычислений. Приводят основные положительные и отрицательные стороны применения технологии Intel Threading Building Blocks, класса Thread C++, технологии NVIDIA CUDA. Решается задача поиска оптимального пути методом Дейкстры. Для усложнения задачи в качестве входных параметров используются шесть показаний датчиков (данные получают с внешних устройств) и двадцать три коэффициента меняющихся интервално. Проводятся сравнительные оценки работы указанных технологий.

Ключевые слова: параллельные алгоритмы, управление, анализ, синтез, датчики

I. ВВЕДЕНИЕ

В мире современных информационных технологии особое место занимают исследования направленные на развитие высокоскоростных методов обработки данных. Поиск оптимальных маршрутов, исследования направленные на выявления закономерностей процессов и явлений, расчеты оптимальных и робастных алгоритмов и т. д., это далеко не полный перечень направлений деятельности «мягких» вычислений. Рассмотрим одно из таких задач.

В условиях постоянно меняющейся инфраструктуры и динамики застройки городских кварталов, остро встает вопрос поиска оптимальных путей перевозки грузов. В данный момент эта задача решается следующим образом. Сервер собирает и обрабатывает поступающие на него данные о текущем дорожном состоянии и выдает оптимальные алгоритмы движения. Как правило, такие данные делятся на два вида: статические и динамические. К статическим типам данных относят: класс дорожного покрытия, ширина полос, техническое состояние, ограничение скорости движения и т. д. К динамическим данным относят наличие пробок, обвалов, повреждения дорожного покрытия, обвалы. Если говорить о первом типе данных, то он относительно стабилен. Второй же постоянно меняется и очень зависит от условий эксплуатации и отрасли применения. Если рассматривать с экономической точки зрения [1, 2], то в основе этих задач лежит снижение себестоимости перевозки. Но зачастую в ущерб экономики могут стать задачи экологии. В некоторых странах эту проблему решают с помощью изменения вида топлива, заменяя его, например, на биодизельное [3, 4]. Другие страны создают разветвленную дорожную сеть,

обеспечивающую нужную логистику. Оба пути имеют как свои преимущества, так и свои недостатки. Решение логистической задачи за счет перехода на биодизельное топливо очень перспективно. Однако оно требует больших экономических затрат для получения сырья и технологической цепочки для её последующей переработки в топливо. В странах обладающих данной технологией и достаточным количеством сырья, это перспективная технология. Но для стран, где технология отсутствует, такой переход невозможен. Решение логистической задачи с помощью расширения сети дорог так же перспективно. Но оказывает отрицательное влияние на экологию. В особенности для промышленно развитых регионов, осуществляющую промышленную добычу угля [5, 6, 7], плавку металлогенных руд [8], добычу нефти и газа [9] извлечению минеральной и артезианской воды [10–15]. В указанных регионах ухудшение экологической обстановки приводит к росту заболеваний и ухудшению психологического состояния человека в целом [16, 17]. Поэтому решение задачи поиска оптимального пути с учетом как экономических, так и экологических факторов является важной и наукоемкой задачей.

II. ПОСТАНОВКА ЗАДАЧИ

Поставим задачу разработки программного модуля на основе алгоритма Дейкстры, для решения задачи перевозки грузов. При разработке алгоритма будем учитывать постоянную динамическую связь с источниками данных (датчики, системы обратной связи). Для обеспечения скорости обработки данных будем использовать параллельные алгоритмы. С целью сравнения существующих технологий, приведем ряд из них: Intel Threading Building Blocks, класс Thread C++, технологию NVIDIA CUDA. Ввиду ограниченности объема статьи рассмотрим краткое описание технологии и его применение.

III. INTEL ® THREADING BUILDING BLOCKS (INTEL ® TBB)

Intel Threaded Building Blocks (TBB) – это набор инструментов с открытым исходным кодом для поддержки масштабируемого параллельного программирования на C++. Это инструментарий более высокого уровня, чем потоки C++, поэтому, его можно использовать с любым компилятором поддерживающих ISO C + +. Библиотека отличается от типичных Threading пакетов наличием 15 Intel ® Threading блоков,

которые дают возможность создавать логические потоки, вместо потоков данных. В Intel® Threading Building Blocks библиотеки времени выполнения автоматически сопоставляют логический параллелизм. Кроме того, она полностью поддерживает вложенный параллелизм. Многие библиотечные интерфейсы используют обобщенное программирование, в котором интерфейсы определяются требованиями по видам, а не конкретных типов.

Таким образом, программирование с использованием Intel® Threading Building Blocks (Intel® TBB) заключается в интеграции особых команд идентификаторов в точки программного кода (точки входа), определяемые программистом. Таким образом, программисту не нужны большие знания математики. Ему достаточно знать с какого момента нужно начать параллельный алгоритм и установить в него нужную метку.

Для решения нашей задачи метки были установлены на прямой и обратный обход графа. Таким образом, было создано три потока данных, два из которых осуществляли обход графа одновременно. Для обычного потока данных это выглядит следующим образом

```
void SerialApplyFoo( float a[], size_t n )
{ for (size_t i=0; i<array_size; ++i)
  { my_func(x[i]);
  } }
```

Код создания потока данных в параллельном виде представлен ниже:

```
class ApplyFunc {double *const my_x;
public:
void operator()(const tbb::blocked_range<size_t>& r)
const {
double *x = my_x;
for(size_t i=r.begin(); i!=r.end(); ++i)
my_func(x[i]);
}
ApplyFunc(double x[]):
my_x{x};
}
```

Таким образом, было произведено разбиение каждой из ветвей графа.

IV. THREAD C++

Класс и библиотека, первоначально созданная для языка C. С выходом стандарта C++ 11 язык получил библиотеку для многопоточного программирования и подходящую модель памяти. Эта библиотека содержит такие строительные блоки, как атомарные переменные, классы для потоков, двоичных семафоров и переменных условия. Они составляют фундамент, на котором в будущих версиях стандарта – например, C++ 20 и C++ 23 – станет возможным определить абстракции более высокого уровня. Однако и в стандарте C++ 11 уже присутствует понятие задания, которое обеспечивает более высокий уровень абстрагирования по сравнению с перечисленными базовыми строительными блоками.

ТАБЛИЦА I. Эволюция языка C++ в контексте использования параллельных алгоритмов [18]

C++11	C++14	C++17	C++20	C++23
2011	2014	2017	2022	2023
Модель памяти	Блокировка и чтения-записи	Поддержка параллельных вычислений в стандартных алгоритмах	Атомарные умные указатели	Исполнители
Задания			Потоки с ожиданием std::jthread	Векторы с параллельной обработкой
Переменные условия			Защелки и барьеры	Блоки заданий
Двоичные семафоры и блокировка			Семафоры общего вида	Транзакционная память
Локальные данные потока			Сопрограммы (coroutine)	Расширения std::future
Потоки				
Атомарные переменные				

С появлением стандарта C++ 17 поддержка параллельного программирования языком C++ значительно расширилась, особенно за счёт параллельных алгоритмов. Стандарты C++ 11 и C++ 14 содержали лишь простейшие строительные блоки для создания параллельных программ. Эти инструменты были удобны для разработки библиотек или каркасов, но не для разработки приложений. Ключевой особенностью C++ 20 и C++ 23 является появление исполнителей. Исполнитель (executor) содержит набор правил касательно того, где, когда и как выполнять вызываемый объект. Они представляют собой основные блоки, из которых строится выполнение программы, и определяют, должен ли тот или иной код выполняться в произвольном потоке, в пуле потоков или даже в едином потоке без распараллеливания. От них зависят расширения фьючерсов, расширения для работы сетью N4734, а также параллельные алгоритмы из стандартной библиотеки; другие средства параллельного программирования из стандартов C++ 20/23, такие как защелки, барьеры, сопрограммы, транзакционная память и блоки заданий, также смогут использовать исполнители.

Гримм Р. В своей книге [18] очень подробно описал возможности применения Исполнителей для задач применения параллельных алгоритмов. Основываясь на ней, был разработан модифицированный метод Дейкстры. Особенностью, которого является жесткая привязка выполнения каждого потока данных на независимых ядрах процессора.

V. NVIDIA CUDA

Данная технология разработана компанией NVIDIA и заключается в использовании математически сложных алгоритмов на вычислительных ядрах видеокарты. Графические ядра видеокарты, GPU по своей сути представляют собой множество арифметико-логических процессоров, для которых выделен небольшой объем видеопамяти с КЭШем первого и второго уровня. В виду высокой частоты работы видеокарты, такие вычисления будут происходить значительно быстрее, чем на процессоре компьютера, CPU. Но в программировании таких алгоритмов есть ряд сложностей. В частности компьютер не может напрямую работать с ядром

графического процессора. В этом случае необходимо загрузить данные для расчета на GPU, произвести вычисления, а затем выгрузить данные обратно. Для задач расчета конкретного значения данная технология представляется оптимальной. В случае задач с множеством побочных расчетов или вывода большого объема данных возникает проблема. Объем памяти видеокарты. В случае компьютера, память можно легко увеличить. В случае с GPU этого сделать невозможно.

В решении алгоритма Дейкстры задача решалась аналогично технологии ТВВ с той разницей, что указанный алгоритм модифицирован загрузкой данных на GPU.

VI. СРАВНЕНИЕ РЕЗУЛЬТАТОВ

Для сравнения работы различных алгоритмов проведем исследование с одинаковым набором данных. В качестве исходных данных установим следующие характеристики: городская застройка с разветвленной структурой дорог, асфальтированное дорожное покрытие, умеренный уровень загрузки и т. д. Всего установлено двадцать три весовых коэффициента. В качестве источников данных установим: загруженность дорог (генерируется случайным образом с интервалом три минуты), температура и влажность дорожного покрытия (генерируется случайным образом с интервалом двадцать минут), авария на дороге (генерируется случайным образом с интервалом один час) и др. Всего шесть параметров. Время моделирования десять часов. Сводное время моделирования представлено в табл. 2.

ТАБЛИЦА II. СРАВНЕНИЕ ЗАТРАТ ВРЕМЕНИ НА РАБОТУ ПРОГРАММНЫХ МОДУЛЕЙ (СЕКУНДЫ)

Номер эксперимента	Технология параллельных вычислений		
	ТВВ	С++	CUDA
6	1700	2700	3000
5	1300	2300	2500
4	1000	2000	2000
3	800	1800	1500
2	700	1300	100
1	500	1000	50

В экспериментах, представленных в табл. 2, номер эксперимента соответствует количеству динамических входных характеристик, где №6 соответствует 6 входным параметрам.

VII. АНАЛИЗ РЕЗУЛЬТАТОВ

Полученные экспериментальные исследования позволяют утверждать, что использование параллельных алгоритмов имеет ряд преимуществ и недостатков. Если обратить внимание на скорость выполнения расчета технологии ТВВ и С++, в зависимости от количества динамических параметров, то можно увидеть практически равномерное уменьшение. В тоже время как у технологии CUDA она носит скачкообразный характер с явным скоростным перевесом. Это связано с тем, что для работы технологии необходимо постоянно перебрасывать данные с CPU на GPU. На что расходуется значительный временной ресурс. В тоже время, когда данные полностью загружены, расчет происходит значительно быстрее, чем при других

технологиях. Из этого можно сделать вывод об эффективности использования технологии CUDA для задач моделирования и расчета систем не реального времени. В случае задач, где имеет место быть постоянное поступление данных с внешнего носителя, более целесообразно применять технологию ТВВ. Однако, данное утверждение не совсем точно. Данная технология предназначена в большей мере для процессоров компании Intel. В тоже время данная технология позволяет выбрать оптимальный алгоритм. В случае, когда аппаратная часть состоит из нескольких процессоров или ядер, набор функции позволяет, определив количество ядер, создать оптимальное количество потоков. В тоже время как технологии С++ могут создать чрезмерное количество потоков данных. Что перегрузит компьютер.

VIII. ВЫВОДЫ

Современные компьютерные технологии создают огромное количество технологий. Выбор технологии программирования зависит опыта программиста и возможностей [19, 20]. Проведенные исследования показали, что в зависимости от задачи стоящей перед разработчиком необходимо проводить глубокий анализ потоков данных. Исходя из этого анализа, выбирать необходимую технологию. Таким образом, можно сформулировать ряд утверждений:

1. Технология CUDA эффективна в случае монотонного расчета заранее фиксированного объема данных. В случае переизбытка данных или переполнения внутренней памяти видеоадаптера необходима постоянная выгрузка данных на компьютер. Что значительно увеличивает время выполнения алгоритма.
2. Технология Threading Building Blocks от компании Intel эффективна в случае и динамических данных, позволяет закреплять выполнение определенного потока данных за конкретным ядром процесса. Многофункциональна. Но значительно уступает по скорости технологии CUDA. Ограничена в своей работоспособности видеокартами от компании NVIDIA.
3. Библиотеки параллельных алгоритмов интегрированных в С++. Отличаются своей универсальностью. Позволяют быть использованными для любых задач.

Таким образом, в рамках данного исследования получены численные характеристики работы алгоритмов при одинаковых начальных условиях.

СПИСОК ЛИТЕРАТУРЫ

- [1] Khaikin M., Shabalov M., Ivanova D., Shapiro N.A. Possible effects of economy digitalization processes on Russian mining industry from economic theory point of view. In "Advances in Raw Material Industries for Sustainable Development Goals" / Litvinenko (Ed) 2021 Taylor & Francis Group, London, ISBN 978-0-367-75881-32021, pp. 481–491.
- [2] Marinina O., Kirsanova N., Nevskaya M. Circular Economy Models in Industry: Developing a Conceptual Framework // Energies 2022, 15, 9376. <https://doi.org/10.3390/en15249376>
- [3] Ereemeeva, A.M., Ilyashenko, I.S., Korshunov, G.I. The possibility of application of bioadditives to diesel fuel at mining enterprises //

- Mining Informational and Analytical Bulletin, 2022, (10-1), pp. 39–49.
- [4] Kondrasheva N.K., & Ereemeeva A.M. (2023). Production of biodiesel fuel from vegetable raw materials // *Journal of Mining Institute*. <https://doi.org/10.31897/PMI.2022.15>
- [5] Kazanin O.I., Meshkov A.A., Sidorenko A.A. Prospects for development of a technological structure of underground coal mines // *Mining Informational and Analytical Bulletin*, 2022, (6-1), pp. 35–53
- [6] Sidorenko A.A., Sidorenko S.A., Ivanov V.V. Numerical modelling of multiple-seam coal mining at the Taldinskaya-Zapadnaya-2 mine // *ARNP Journal of Engineering and Applied Sciences*. 2021. №5. pp. 568-574.
- [7] Martirosyan A.V., Ilyushin Y.V. The Development of the Toxic and Flammable Gases Concentration Monitoring System for Coalmines // *Energies* 2022, 15, 8917. <https://doi.org/10.3390/en15238917>
- [8] Kukharova T.V., Ilyukhina Y.A., Shestopalov M.Y. Development of a Methodology for Controlling the Process of Heating Metal Blanks in a Methodical Furnace // *Proceedings of the 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2022*, 2022, pp. 718–721.
- [9] Ilyushin Y.V. Development of a Process Control System for the Production of High-Paraffin Oil // *Energies* 2022, 15, 6462. <https://doi.org/10.3390/en15176462>
- [10] Pershin I.M., Kukharova T.V., Tsapleva V.V. Designing of distributed systems of hydrolithosphere processes parameters control for the efficient extraction of hydromineral raw materials // *Journal of Physics: Conference Series*, 2021, 1728(1), 012017
- [11] Martirosyan A.V., Kukharova T.V., Fedorov M.S. Research of the hydrogeological objects' connection peculiarities // *Proceedings of 2021 4th International Conference on Control in Technical Systems, CTS 2021*, 2021, pp. 34–38.
- [12] Martirosyan A.V., Martirosyan K.V., Mir-Amal A.M., Chernyshev A.B. Assessment of a Hydrogeological Object's Distributed Control System Stability // *Proceedings of the 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2022*, 2022, pp. 768–771.
- [13] Asadulagi M.M., Ioskov G.V., Tronina E.V. Synthesis of Lumped and Distributed Controllers for Control System of Hydrodynamic Process // *2019 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2019*, 2019, 8933859
- [14] Asadulagi M.M., Vasilkov O.S. The Use of Distributed and Lumped Type Controllers for the Hydro-lithospheric Process Control System of the Kislovodskoye Field // *Proceedings of 2019 3rd International Conference on Control in Technical Systems, CTS 2019*, 2019, pp. 7–10, 8973272
- [15] Asadulagi M.M., Pervukhin D.A. Stochastic control system of hydrodynamic processes in aquifers Innovation-Based Development of the Mineral // *Resources Sector: Challenges and Prospects – 11th conference of the Russian-German Raw Materials*, 2018, 2019, pp. 175–185.
- [16] Kukharova T.V., Utkin V.A., Boev I.V. Observation and Prediction Systems Modeling for Human Mental State // *2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018*, 2018, 8602831
- [17] Kukharova T.V., Utkin V.A., Pershin I.M. Modeling of a Decision Support System for a Psychiatrist Based on the Dynamics of Electrical Conductivity Parameters // *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2021*, 2021, pp. 975–978, 9396273
- [18] Гримм Р. Параллельное программирование на современном языке C++ / пер. с англ. В.Ю. Винника. М.: ДМК Пресс, 2022. 618 с.
- [19] Ilyushin Y., Mokeev A. Technical realization of the task of controlling the temperature field of a tunnel furnace of a conveyor type // *(2017) International Journal of Applied Engineering Research*, 12 (8), pp. 1500–1510.
- [20] Ilyushin Y.V., Afanasieva O.V. Synthesis of a distributed control system (2016) // *International Journal of Control Theory and Applications*, 9 (30), pp. 41–60.