

ПРИМЕНЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКИ В ЭКСПЕРИМЕНТЕ

УДК 519.683.8

МОДИФИКАЦИЯ МЕТОДА МНОГОКРАТНОЙ ВЗАИМНОЙ  
СИНХРОНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ  
ПОТОКОВ ДЛЯ ПРОЦЕССОРОВ С ТЕХНОЛОГИЕЙ  
HYPERTHREADING

©2026 г. С. Ю. Ксенофонтов<sup>a,\*</sup>

<sup>a</sup> Институт прикладной физики им. А.В. Гапонова-Грехова  
Российской академии наук

Россия, 603950, Нижний Новгород, ул. Ульянова, 46

\*e-mail: xen@ipfran.ru

Поступила в редакцию 31.07.2025 г.

После доработки 22.09.2025 г.

Принята к публикации 02.10.2025 г.

Предложена модификация метода многократной взаимной синхронизации параллельных вычислительных потоков, которая применялась в системах спектральной оптической когерентной томографии под управлением вычислительных систем с центральным процессором с технологией HyperThreading. Данная модификация обеспечивает равномерную вычислительную загрузку всех задействованных логических процессоров и снижение пиковых показателей загрузки. За счет этого была продемонстрирована устойчивая работоспособность вычислительного метода в условиях решения томографических задач различной сложности с применением мультискалярных процессоров разных поколений, разной мощности и разных производителей.

## 1. ВВЕДЕНИЕ

Описанные в этой статье методы организации параллельных вычислительных потоков применялись автором в программно-алгоритмическом обеспечении приборов спектральной оптической когерентной томографии (ОКТ) [1], разработанных в ИПФ РАН.

Основное применение ОКТ – визуализация внутренней структуры биологических тканей в рамках различных биомедицинских исследований и диагностических процедур. ОКТ-технология основана на зондировании исследуемого объекта низкокогерентным

оптическим излучением ближнего инфракрасного диапазона. В спектральной ОКТ используется анализ спектра оптического сигнала, который является результатом интерференции опорного излучения и излучения, рассеянного в обратном направлении неоднородностями зондируемой ткани [2]. Таким образом математически синтезируется изображение томографического среза. Интенсивность пикселей этого изображения пропорциональна пространственному распределению коэффициента обратного рассеяния внутри исследуемого объекта. Метод ОКТ обладает высоким пространственным разрешением (до нескольких микрометров) и малоинвазивен за счет низкой мощности зондирующего излучения.

Основная особенность ОКТ-систем, для которых применялись представленные в этой статье вычислительные методы, заключается в том, что пользователь определяет и удерживает (фиксирует) текущее положение сканирующего зонда, исходя из текущего ОКТ-изображения. Другими словами, эти ОКТ-системы должны обеспечивать получение в реальном времени достоверной информации о микромасштабной структуре исследуемого объекта, а также нацеливание зонда в условиях физиологических движений биоткани, т. е. пользователю должна быть предоставлена в реальном времени наглядная информация о результатах его манипуляций, что должно обеспечить интерактивность его действий. Таким образом, формулируемая здесь задача относится к классу задач жесткого реального времени.

Характерной конструктивной особенностью этих ОКТ-систем является то, что они разработаны в виде аппаратно-программных комплексов, т. е. управление ОКТ-системой, синтез и визуализация непрерывного потока ОКТ-изображений выполняются персональным компьютером в условиях функционирования операционной системы общего назначения. Персональный компьютер связан с прочей аппаратной частью ОКТ-системы посредством специально разработанной системы сбора данных и управления [3–5]. Для связи с компьютером в этом случае используется интерфейс USB. Так обеспечивается универсальность этого конструктивного решения, поскольку с точки зрения программно-алгоритмического обеспечения не имеет значения, какой компьютер применяется (настольный, мобильный, или встроенный), однако применение интерфейса USB влечет дополнительные требования к применяемым алгоритмам.

Вычислительные процедуры синтеза спектрального ОКТ-изображения – это сложный комплекс математических манипуляций, таких как преобразования Фурье, пространственная фильтрация и манипуляции с фазой комплексных данных. Подавляющее большинство известных спектральных ОКТ-систем сторонних разработчиков и производителей используют для решения этих задач вычислительную мощность графических процессоров (например [6, 7]). Но в ряде случаев такой подход неприемлем или нецелесообразен.

Например, это может быть связано с требованием минимизации себестоимости изделия или с требованием компактности и мобильности. Кроме того, в компьютерах, сертифицированных для применения в условиях хирургических операционных, как правило, отсутствует дискретная графика. Это связано с тем, что в условиях биологически-чистых помещений применение оборудования с принудительным воздушным охлаждением нежелательно. Поэтому в нашем случае используется вычислительная мощность только одного многоядерного центрального процессора мультисклярной архитектуры.

Особенность функционирования интерфейса USB в условиях операционных систем общего назначения состоит в том, что на уровне драйвера USB-устройства передача данных в компьютер всегда производится мелкими порциями (например, по 256 байт для USB 2.0 и по 1024 байта для USB 3.0). При этом в рамках функционирования операционной системы общего назначения на каждую такую транзакцию центральный процессор так или иначе отвлекается. Отметим, что исходные данные одного кадра ОКТ-изображения (далее В-скана) имеют размер до нескольких мегабайт. Кроме того, применяемые алгоритмы синтеза В-скана в спектральной ОКТ предполагают обработку всех исходных данных, составляющих этот В-скан. Другими словами, В-скан нельзя построить по частям. Таким образом, в случае чрезмерной загрузки центрального процессора вычислительными процедурами синтеза и визуализации ОКТ-изображений в операционной системе общего назначения неизбежны задержки в передаче данных по USB-каналу. Следует обратить внимание на то, что применяемые нами системы сбора данных и управления имеют пропускную способность, близкую к предельной гарантированной пропускной способности используемых в них USB-контроллеров. Эти системы рассчитаны на непрерывный асинхронный сбор данных, т. е. сбор данных осуществляется непрерывно и проводится параллельно с вычислительными процедурами синтеза непрерывного потока В-сканов. В таких условиях задержки в передаче данных по USB-каналу чреваты потерями данных, и эта проблема не может быть решена дополнительной аппаратной буферизацией.

Исходя из этого, мы можем сформулировать одно из основных требований к применяемым методам организации параллельных вычислительных потоков. Оно состоит в том, что для устойчивой работы такой ОКТ-системы вычислительная загрузка ни одного из процессорных ядер не должна достигать 100%. Это необходимо, чтобы у менеджера задач операционной системы всегда имелась возможность выделить необходимое количество процессорных ресурсов для функционирования USB-драйвера и прочих системных сервисов.

Логично предположить, что равномерное распределение вычислительной загрузки процессорных ядер в определенных условиях позволяет это обеспечить. Описанию таких методов параллельных вычислений посвящается настоящая статья.

## 2. МЕТОД МНОГОКРАТНОЙ ВЗАИМНОЙ СИНХРОНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПОТОКОВ

Ранее для решения поставленной выше задачи автором был предложен метод многократной взаимной синхронизации параллельных вычислительных потоков [8]. Общая концепция метода состоит в том, что на начальном этапе инициализации основной пользовательской программы создается  $N$  приостановленных вычислительных потоков.  $N$  равно числу доступных логических процессоров. Каждый из этих  $N$  вычислительных потоков жестко ассоциируется с одним из доступных логических процессоров. Таким образом, каждый из этих потоков будет выполняться в отдельном логическом процессоре. Эти потоки выполняют одну и ту же процедуру, показанную на рис. 1. Кроме того, создается еще один приостановленный асинхронный вычислительный поток, предназначенный для процедуры непосредственного отображения готового В-скана на мониторе. Этот поток не имеет жесткой ассоциации с конкретным логическим процессором.

**Рис. 1.**

На рис. 1 показано, что общий алгоритм разбит на  $K$  этапов. Это разбиение необходимо в тех случаях, когда каждый следующий этап зависит от результатов предыдущего. В противном случае  $K = 1$ . Эти этапы разделены процедурами сообщения о готовности соответствующего блока вычислений и процедурами ожидания общей готовности.

Внутри каждого этапа выполняется  $M$  процедур, доступных для параллельного вычисления. Данный метод основан на том, что, в частности, для алгоритмов спектральной ОКТ справедливо утверждение, что в большинстве случаев можно реализовать обработку промежуточных данных так, что  $M > N$  (в случае использования центрального процессора настольного или мобильного типа). Предложенный метод организован так, что на каждом из  $K$  этапов одновременно выполняется  $N$  из  $M$  алгоритмически одинаковых процедур, обозначенных на блок-схеме рис. 1 как “Блок вычислений  $n$ ”. Это происходит за счет особенности организации основных циклов обработки. Они обозначены на блок-схеме рис. 1 как “Шаг на  $N$ ”. На языке C++ организация такого цикла может быть записана в виде

```
for(int i = P; i < M; i += N) {...},
```

где  $P$  – номер текущего вычислительного потока от 0 до  $N-1$ , или, другими словами, номер логического процессора, используемого этим потоком.

Практическое использование метода многократной взаимной синхронизации параллельных вычислительных потоков продемонстрировало возможность равномерного распределения вычислительной нагрузки на все используемые процессорные ядра. Этот

фактор позволяет уменьшить максимальную пиковую нагрузку для каждого процессорного ядра, что способствует стабильной работе асинхронных процедур сбора данных и предотвращает потери данных и соответствующие аварийные ситуации.

### 3. ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ ПРИ ИСПОЛЬЗОВАНИИ ПРОЦЕССОРОВ С ТЕХНОЛОГИЕЙ HyperThreading

Описанный выше метод многократной взаимной синхронизации параллельных вычислительных потоков гарантирует однозначный положительный результат в условиях использования центрального процессора, в котором число логических процессоров равно числу процессорных ядер. Однако это условие не выполняется для мультискэлярных процессоров с технологией HyperThreading [9].

Технология HyperThreading удваивает количество логических процессоров, доступных для параллельного использования, по сравнению с количеством ядер центрального процессора. Суть данной технологии состоит в том, что каждое процессорное ядро содержит избыточное количество виртуальных регистров общего назначения, два конвейера, два контроллера прерываний, и это ядро может хранить два состояния процессора. Таким образом, некоторые не задействованные в данный момент ресурсы могут выполнять действия параллельно другим текущим действиям.

Эта технология вполне эффективна в условиях обычного функционирования многозадачной операционной системы общего назначения, когда одновременно выполняются несколько разных пользовательских приложений и системных сервисов. В этом случае вероятность того, что в данный момент задействуются одинаковые ресурсы процессорного ядра, невелика. Однако в случае работы алгоритма параллельных вычислений, описанного выше, одновременно выполняются процедуры, которые одинаковы с алгоритмической точки зрения, и при использовании технологии HyperThreading неизбежна постоянная конкуренция между разными вычислительными потоками за доступ к ресурсам процессора.

Для решения таких проблем был проведен ряд экспериментов с применением компьютеров на базе различных процессоров и нескольких модификаций ОКТ-систем с различными степенями сложности вычислительных алгоритмов. В результате была выявлена некоторая закономерность, которая может быть проиллюстрирована следующим образом.

На рис. 2 приведена хронология загрузки логических процессоров у центрального процессора Core i3-12100 с отключенным режимом HyperThreading в процессе работы ОКТ-системы. Здесь наглядно продемонстрировано обеспечение равномерного распределения вычислительной нагрузки при помощи метода многократной взаимной синхронизации

параллельных вычислительных потоков. В этом случае число логических процессоров совпадает с числом процессорных ядер. Стоит обратить внимание, что средняя загрузка процессора в этом случае превышает 61%.

**Рис. 2.**

На рис. 3 продемонстрирована загрузка логических процессоров того же центрального процессора, но с включенным режимом HyperThreading. В этом случае число логических процессоров увеличилось вдвое. Применялась та же ОКТ-система с аналогичной организацией параллельных вычислений. Стоит отметить, что эта программа управления ОКТ-системой автоматически распределяет вычислительные потоки на все доступные логические процессоры. В этом случае средняя загрузка процессора составила примерно 46–47%. Однако вычислительная загрузка логических процессоров стала неравномерной, что является следствием конкуренции алгоритмически-одинаковых процедур за ресурсы процессорного ядра.

**Рис. 3.**

Для решения этой проблемы была предложена и реализована следующая модификация описанного выше метода. Основная идея состоит в том, чтобы ассоциировать с каждым логическим процессором не один вычислительный поток, а несколько. Тогда параллельное выполнение этих нескольких вычислительных потоков будет сдвинуто во времени друг относительно друга менеджером задач операционной системы. Таким образом, появляется возможность более равномерного распределения вычислительной загрузки логических процессоров в режиме HyperThreading.

Такая модификация была осуществлена следующим образом. Концептуальная блок-схема вычислительных потоков осталась такой же, как на рис. 1, но было принято решение всегда создавать 32 потока (то есть  $N = 32$ ). В случае наличия 4 логических процессоров с каждым из них будет ассоциировано 8 вычислительных потоков, в случае 8 процессоров – 4 потока и т.д. Результат работы такого метода продемонстрирован на рис. 4. Наблюдается в значительной степени равномерное распределение вычислительной нагрузки. Средняя загрузка процессора составила около 47%. Практика показывает, что средняя загрузка процессора увеличивается в этом случае незначительно, в пределах 1–2%.

**Рис. 4.**

Практическую значимость такого метода можно проиллюстрировать тем фактом, что, например, при применении процессора Core i3-12100T (менее производительного, но с

меньшим тепловыделением) с такой же ОКТ-системой представленная модификация метода организации параллельных вычислений обеспечивает устойчивую работоспособность. При этом немодифицированный метод дает сбои в работе программного обеспечения, так как иногда пиковая загрузка некоторых логических процессоров достигает 100%.

#### 4. ЗАКЛЮЧЕНИЕ

Описанная в данной работе модификация метода многократной взаимной синхронизации параллельных вычислительных потоков продемонстрировала свою работоспособность в условиях задач различной сложности с применением центральных процессоров разных поколений, разной мощности и разных производителей.

Эффективность этого метода позволила практически реализовать некоторые дополнительные модальности ОКТ. В частности, к таким модальностям относятся визуализация микрокапиллярного кровотока [10] и компрессионная эластография [11].

Дополнительная возможность значительного усложнения применяемых ОКТ-алгоритмов обеспечила значительное улучшение качества структурных ОКТ-изображений за счет применения различных методов подавления артефактов спектральной ОКТ [12–18].

#### ФИНАНСИРОВАНИЕ РАБОТЫ

Работа выполнена в рамках государственного задания Института прикладной физики им. А.В. Гапонова-Грехова РАН (проект № FFUF-2024-0029).

#### СПИСОК ЛИТЕРАТУРЫ

1. Optical coherence tomography: Technology and applications / Ed. by W. Drexler, J.G. Fujimoto. Springer, 2015. <http://doi.org/10.1007/978-3-319-06419-2>
2. Fercher A.F., Hitzenberger C.K., Kamp G., El-Zaiat S.Y. // Opt. Commun. 1995. V. 117. № 1-2. P. 43. [http://doi.org/10.1016/0030-4018\(95\)00119-S](http://doi.org/10.1016/0030-4018(95)00119-S)
3. Геликонов В.М., Геликонов Г.В., Терпелов Д.А., Шилягин П.А. // ПТЭ. 2012. № 3. С. 100. <https://www.elibrary.ru/item.asp?id=17726204>
4. Терпелов Д.А., Ксенофонтов С.Ю., Геликонов Г.В., Геликонов В.М., Шилягин П.А. // ПТЭ. 2017. № 6. С. 94. <http://doi.org/10.7868/S0032816217060143>
5. Ксенофонтов С.Ю., Купаев А.В., Василенкова Т.В., Терпелов Д.А., Шилягин П.А., Mouseev A.A., Геликонов Г.В. // ПТЭ. 2021. № 5. С. 131. <http://doi.org/10.31857/S0032816221040224>

6. Majumdar A., Allam N., Zabel W.J., Demidov V., Flueraru C., Vitkin A. // Sci. Rep. 2022. V. 12. P. 13995. <http://doi.org/10.1038/s41598-022-18393-4>
7. Kim W., Long R., Yang Z., Oghalai J.S., Applegate B.E. // J. Biomed. Opt. 2024. V. 29 № 8. P. 086005. <http://doi.org/10.1117/1.JBO.29.8.086005>
8. Ксенофонтов С.Ю. // ПТЭ. 2019. № 3. С. 17. <http://doi.org/10.1134/S0032816219030078>
9. Marr D.T., Binns F., Hill D.L., Hinton G., Koufaty D.A., Miller J.A., Upton M. // Intel Technology J. 2002. V. 6. № 1. P. 4.
10. Maslennikova A.V., Sirotkina M.A., Moiseev A.A. et al. // Sci. Rep. 2017. V. 7. P. 16505. <http://doi.org/10.1038/s41598-017-16823-2>
11. Zaitsev V.Y., Ksenofontov S.Y., Sovetsky A.A., Matveyev A.L., Matveev L.A., Zykov A.A., Gelikonov G.V. // Photonics. 2021. V. 8. № 12. P. 527. <http://doi.org/10.3390/photonics8120527>
12. Ксенофонтов С.Ю., Терпелов Д.А., Геликонов Г.В., Шилягин П.А., Геликонов В.М. // Изв. вузов. Радиофизика. 2019. Т. 62. № 2. С. 167. <https://www.elibrary.ru/item.asp?id=37749323>
13. Геликонов Г.В., Ксенофонтов С.Ю., Шилягин П.А., Геликонов В.М. // Изв. вузов. Радиофизика. 2019. Т. 62. № 3. С. 252. <https://www.elibrary.ru/item.asp?id=38480973>
14. Ксенофонтов С.Ю., Шилягин П.А., Терпелов Д.А., Новожилов А.А., Геликонов В.М., Геликонов Г.В. // ПТЭ. 2020. № 1. С. 136. <http://doi.org/10.31857/S003281622001005X>
15. Ксенофонтов С.Ю., Мoiseев A.A., Matkivskiy B.A., Shilyagin P.A., Vasilenkova T.B., Геликонов В.М., Геликонов Г.В. // ПТЭ. 2020. № 5. С. 104. <http://doi.org/10.31857/S0032816220040291>
16. Ksenofontov S.Y., Shilyagin P.A., Terpelov D.A., Shabanov D.V., Gelikonov V.M., Gelikonov G.V. // Front. Optoelectron. 2020. V. 13 № 4. P. 393. <http://doi.org/10.1007/S12200-019-0951-0>
17. Ksenofontov S.Y., Shilyagin P.A., Gelikonov V.M., Gelikonov G.V. // Photonics. 2023. V. 10 № 7. P. 736. <http://doi.org/10.3390/photonics10070736>
18. Ксенофонтов С.Ю., Шилягин П.А., Терпелов Д.А., Шабанов Д.В., Геликонов В.М., Геликонов Г.В. // ПТЭ. 2023. № 6. С. 154. <http://doi.org/10.31857/S0032816223050312>

## ПОДПИСИ К РИСУНКАМ

**Рис. 1.** Блок-схема потоков параллельных вычислений с многократной синхронизацией.

**Рис. 2.** Диаграмма загрузки логических процессоров в условиях применения метода многократной взаимной синхронизации параллельных вычислительных потоков при отключенном режиме HyperThreading .

**Рис. 3.** Диаграмма загрузки логических процессоров в условиях применения метода многократной взаимной синхронизации параллельных вычислительных потоков при включенном режиме HyperThreading.

**Рис. 4.** Диаграмма загрузки логических процессоров в условиях применения модифицированного метода многократной взаимной синхронизации параллельных вычислительных потоков при включенном режиме HyperThreading.

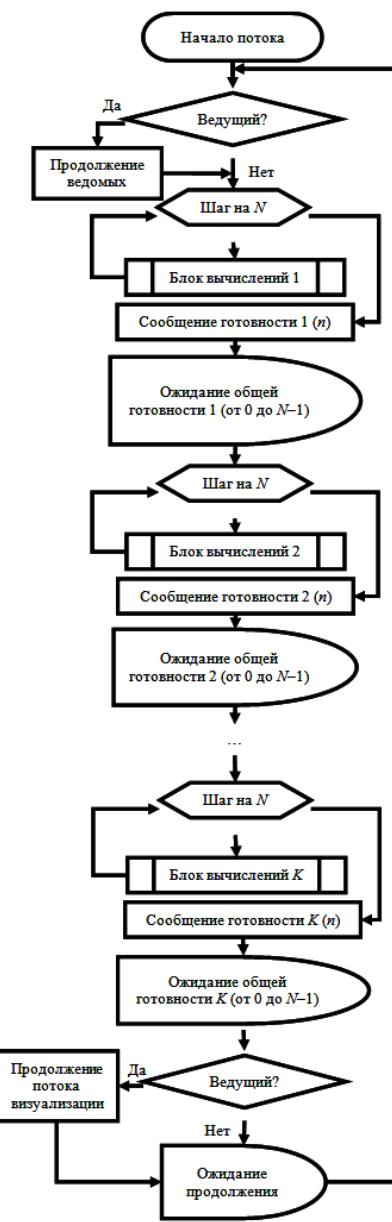
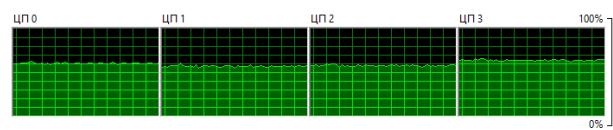
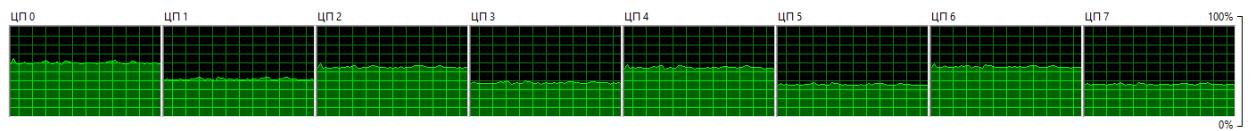


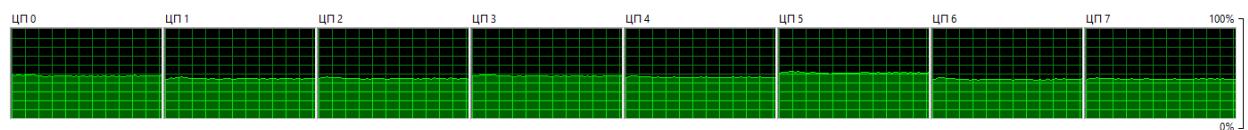
Рис. 1



**Рис. 2**



**Рис. 3**



**Рис. 4**