

Ссылка для цитирования этой статьи:

Борисов А.А., Ростоцкий М.В., Шорохов К.Д., Кучук М.И., Гардаш В.В. Основные теоретические и практические подходы в информатике и вычислительных технологиях, используемые при разработке программно-информационных систем // Human Progress. 2024. Том 10, Вып. 6. URL: http://progress-human.com/images/2024/Tom10_6/Borisov.pdf DOI 10.46320/2073-4506-2024-6a-2.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ И ПРАКТИЧЕСКИЕ ПОДХОДЫ В ИНФОРМАТИКЕ И ВЫЧИСЛИТЕЛЬНЫХ ТЕХНОЛОГИЯХ, ИСПОЛЬЗУЕМЫЕ ПРИ РАЗРАБОТКЕ ПРОГРАММНО-ИНФОРМАЦИОННЫХ СИСТЕМ

Борисов Артём Андреевич

студент Дальневосточного федерального университета
г. Владивосток, Российская Федерация

Ростоцкий Максим Викторович

студент Дальневосточного федерального университета
г. Владивосток, Российская Федерация

Шорохов Константин Дмитриевич

студент Дальневосточного федерального университета
г. Владивосток, Российская Федерация

Кучук Максим Игоревич

студент Дальневосточного федерального университета
г. Владивосток, Российская Федерация

Гардаш Владислав Викторович

студент Дальневосточного федерального университета
г. Владивосток, Российская Федерация

Аннотация. В данной статье рассматриваются различия между теоретическими и практическими подходами в области вычислительных технологий. Объект исследования – информационная экономика. Предмет исследования – системные элементы информационной экономики. Цель исследования – раскрыть основные теоретические и практические подходы в информатике и вычислительных технологиях, используемые при разработке программно-информационных систем. Теоретические модели, такие как анализ сложности алгоритмов, теория параллельных вычислений и теория сложности, предоставляют фундаментальные принципы для разработки эффективных решений. Однако на практике эти теории сталкиваются с ограничениями, вызванными реальными условиями, такими как доступность

ресурсов, сложность данных и требования к производительности. В статье анализируются ключевые примеры, такие как применение параллельных алгоритмов, машинного обучения, вычислительной геометрии и баз данных, с акцентом на то, как теоретические концепции адаптируются к практическим задачам. В результате исследования выявляются стратегии, позволяющие преодолеть разрыв между теорией и практикой, и подчеркивается важность гибридных подходов в современной информатике.

Ключевые слова: теоретическая информатика, вычислительные технологии, параллельные вычисления, машинное обучение, алгоритмы, инновации, новые технологии.

Введение

Вычислительная экономика представляет собой междисциплинарную исследовательскую область, объединяющую в себе информатику, экономику и науку об управлении, в которой основная сфера деятельности заключается в разработке вычислительных моделей экономических систем. Эта область исследований включает в себя ряд уникальных направлений, а также способствует развитию уже существующих областей экономики, так благодаря вычислительной экономике стало возможным проводить глубокий анализ данных и решать сложные задачи, которые было бы крайне сложно изучить без использования компьютеров и соответствующих численных методов. В области информатики и вычислительных технологий теоретические подходы играют ключевую роль в развитии как фундаментальных, так и прикладных направлений. Одним из центральных аспектов этой области является разработка и анализ алгоритмов, которые позволяют эффективно решать сложные задачи. Алгоритмы можно рассматривать как пошаговые инструкции для решения конкретных задач, где важнейшим параметром является их эффективность. Например, алгоритмы сортировки, такие как сортировка слиянием с временной сложностью $O(n \log n)$, являются базисом для множества приложений, требующих организации больших объемов данных. Современные методы анализа алгоритмов основываются на вычислительных моделях, таких как машина Тьюринга или RAM-модели, где основное внимание уделяется вычислительной сложности.

Основная часть

Важным понятием в теории алгоритмов является асимптотическая сложность, которая позволяет оценить поведение алгоритма на больших объемах данных. Например, в приложениях к большим данным оптимизация сложности алгоритмов позволяет значительно улучшить производительность систем [1]. В реальной практике, где объем данных может измеряться в петабайтах, разница между алгоритмами с $O(n^2)$ и $O(n \log n)$ сложностью

может определять практическую возможность их использования. Это стало особенно актуально с развитием технологий больших данных, где даже небольшое улучшение в эффективности приводит к экономии огромных вычислительных ресурсов и времени.

Теоретические подходы также включают разработку и анализ структур данных, которые обеспечивают эффективное хранение и манипулирование информацией. Например, такие структуры, как двоичные деревья поиска (BST) или хеш-таблицы, широко используются в различных системах для оптимизации операций поиска, вставки и удаления данных. В больших системах, таких как распределенные базы данных или файловые системы, выбор оптимальной структуры данных может существенно повлиять на общую производительность системы. Например, самобалансирующиеся деревья, такие как красно-черные деревья или деревья AVL, позволяют гарантировать, что операции поиска и вставки всегда выполняются за логарифмическое время $O(\log n)$, что критически важно для систем с высокими нагрузками.

Одним из важнейших направлений теоретической информатики является теория сложности, которая исследует границы вычислимости и классифицирует задачи по их вычислительной трудности. Например, одна из наиболее известных проблем в теории сложности — это проблема P vs NP , которая остаётся открытой и является центральной задачей современной теоретической информатики. Вопрос о том, можно ли решить задачи класса NP за полиномиальное время, имеет значительные последствия для многих областей, включая криптографию, оптимизацию и биоинформатику. На практике многие сложные задачи, такие как задача коммивояжера, являются NP -трудными, что делает их решение за полиномиальное время невозможным. Однако благодаря разработке приближенных алгоритмов и эвристик можно получать решения, близкие к оптимальным за разумное время. Например, в задачах логистики, где требуется найти оптимальные маршруты, эвристические методы, такие как алгоритмы ближайшего соседа или генетические алгоритмы, позволяют находить решения, которые, хотя и не оптимальны, но достаточно хороши для практического использования.

С ростом объема данных и сложностью вычислительных задач особое внимание уделяется параллельным вычислениям. Параллельные вычисления позволяют распределять задачи между несколькими процессорами, тем самым значительно ускоряя выполнение сложных программ. Теоретические исследования в этой области сосредоточены на разработке моделей параллельных вычислений, таких как PRAM (Parallel Random Access Machine), а также на создании алгоритмов, которые могут эффективно использовать большое количество процессоров [2]. Например, параллельная сортировка слиянием или параллельные алгоритмы для поиска в графах позволяют значительно сократить время вычислений в

многопроцессорных системах. Важным вопросом здесь является не только разработка эффективных алгоритмов, но и учет накладных расходов на коммуникацию между процессорами. Например, в системе MapReduce, которая активно используется для обработки больших данных, параллельные задачи разбиваются на множество мелких подзадач, что позволяет эффективно использовать ресурсы распределенной системы.

Еще одним важным направлением является вычислительная геометрия, которая занимается разработкой алгоритмов для решения задач, связанных с геометрическими объектами. Алгоритмы, такие как алгоритм выпуклой оболочки или алгоритмы триангуляции, находят широкое применение в таких областях, как компьютерная графика, робототехника и географические информационные системы (ГИС). Например, алгоритм Грэхема, который имеет временную сложность $O(n \log n)$, позволяет эффективно строить выпуклую оболочку множества точек. Это важно при решении задач, связанных с планированием маршрутов или анализом геометрических объектов. В робототехнике вычислительная геометрия используется для построения карт и планирования движения роботов, а в компьютерной графике — для рендеринга и моделирования трёхмерных объектов.

Параллельно с развитием традиционных вычислительных моделей значительное внимание уделяется исследованиям в области квантовых вычислений, которые обещают революционные изменения в решении сложных задач. Квантовые алгоритмы, такие как алгоритм Шора для факторизации чисел или алгоритм Гровера для поиска, показывают, что некоторые задачи, которые в классической теории сложности считаются трудно решаемыми, могут быть решены намного быстрее на квантовых компьютерах. Хотя квантовые вычисления всё ещё находятся на этапе активных исследований, их теоретический потенциал уже сейчас открывает новые горизонты для решения задач в криптографии, моделировании молекул и оптимизации.

Практические применения вычислительных технологий охватывают широкий спектр отраслей, от бизнеса и промышленности до медицины и науки. В последние десятилетия вычислительные технологии стали основой для решения сложных задач, которые требуют обработки больших объемов данных, выполнения сложных математических расчетов и моделирования процессов в реальном времени. Их успешное применение привело к значительным изменениям в таких областях, как анализ данных, искусственный интеллект, биоинформатика, моделирование в инженерии, а также в повседневных аспектах жизни, таких как электронная коммерция и социальные сети.

Одним из наиболее заметных примеров использования вычислительных технологий является анализ больших данных. В условиях современного информационного общества

объемы данных растут экспоненциально, и их обработка требует высокопроизводительных вычислительных ресурсов. Примеры таких данных включают информацию, генерируемую пользователями в социальных сетях, данные с сенсоров интернета вещей (IoT), медицинские изображения и геномные последовательности. Для эффективного анализа этих данных применяются технологии распределенных вычислений, такие как MapReduce и Hadoop, которые позволяют обрабатывать петабайты данных параллельно на тысячах узлов. Эти технологии используются в таких сферах, как маркетинг, финансовый анализ, исследования в биоинформатике и мониторинг сетевой активности.

Искусственный интеллект (ИИ) и машинное обучение также находятся в центре внимания практических применений вычислительных технологий. Системы ИИ используются для автоматизации различных процессов, таких как распознавание изображений, обработка естественного языка, прогнозирование и диагностика [3]. Например, в медицинских исследованиях машинное обучение применяется для анализа больших наборов данных с целью выявления скрытых закономерностей, что позволяет улучшить диагностику заболеваний и разработать новые методы лечения. В таких задачах, как анализ рентгеновских снимков или МРТ, нейронные сети способны обнаруживать аномалии с точностью, сравнимой или превосходящей человеческую. В коммерческой сфере ИИ используется для улучшения взаимодействия с клиентами через чат-боты и системы рекомендации, которые персонализируют предложения на основе анализа поведения пользователей.

В облачных вычислениях вычислительные технологии позволяют предоставлять вычислительные ресурсы и услуги через интернет. Облачные платформы, такие как Amazon Web Services (AWS), Microsoft Azure и Google Cloud, предоставляют возможности для хранения, обработки данных и выполнения приложений без необходимости владеть собственной инфраструктурой. Это дало значительный толчок развитию стартапов и малых предприятий, которые могут использовать ресурсы облачных платформ для создания сложных систем и сервисов с минимальными затратами на инфраструктуру. Одним из примеров практического использования облачных технологий является автоматизация бизнес-процессов с помощью облачных ERP-систем, которые объединяют все аспекты управления предприятием — от финансов и логистики до человеческих ресурсов и маркетинга.

В промышленности и инженерии вычислительные технологии применяются для моделирования и симуляции сложных процессов. Например, в аэрокосмической и автомобильной отраслях широко используются методы вычислительной гидродинамики (CFD) для моделирования потоков воздуха вокруг объектов. Эти симуляции позволяют инженерам тестировать и оптимизировать конструкции без необходимости проведения

дорогостоящих физических испытаний. В химической и биологической инженерии применяются методы молекулярного моделирования, которые позволяют изучать взаимодействия между молекулами и прогнозировать поведение новых материалов или лекарственных препаратов. В совокупности с методами оптимизации эти технологии значительно ускоряют цикл разработки и позволяют получать точные результаты на более ранних этапах исследований.

Биоинформатика является еще одной областью, где вычислительные технологии играют ключевую роль. Геномные данные, которые могут содержать миллионы последовательностей ДНК, требуют значительных вычислительных ресурсов для их анализа и интерпретации. Современные алгоритмы для выравнивания последовательностей, такие как BLAST и FASTA, позволяют ученым находить сходства между последовательностями ДНК или белков, что важно для изучения эволюции, генетических заболеваний и разработки новых лекарственных средств. Вычислительные технологии также используются для моделирования биологических систем, таких как сети метаболических путей, что помогает в исследовании того, как организм реагирует на различные стимулы или лекарства.

Машинное зрение и робототехника также значительно выиграли от развития вычислительных технологий. Машинное зрение — это технология, которая позволяет компьютерам интерпретировать и понимать визуальную информацию из окружающего мира. Это применимо в таких областях, как автоматизация производства, автономные транспортные средства и системы видеонаблюдения. Например, на заводах машины с компьютерным зрением могут контролировать качество продукции в режиме реального времени, выявляя дефекты на конвейерных линиях с высокой скоростью и точностью. В автомобилестроении автономные транспортные средства используют технологии компьютерного зрения и машинного обучения для распознавания объектов на дороге, построения маршрутов и принятия решений в реальном времени.

Еще одной важной областью применения вычислительных технологий является криптография и обеспечение безопасности информации. В условиях цифровой экономики защита данных стала критически важной задачей для банков, правительств и компаний. Развитие методов симметричного и асимметричного шифрования позволило обеспечить безопасность транзакций в интернете, защиту персональных данных и цифровую идентификацию. Такие алгоритмы, как RSA, AES и SHA-256, широко используются для обеспечения конфиденциальности и целостности данных. Внедрение блокчейн-технологий также стало значительным прорывом в обеспечении безопасности транзакций, особенно в финансовом секторе. Блокчейн представляет собой распределённую базу данных, которая

защищена криптографическими методами и обеспечивает прозрачность и неизменность данных.

В современных вычислительных технологиях инструменты и технологии играют ключевую роль, обеспечивая высокую производительность, гибкость и масштабируемость для решения разнообразных задач. Они включают в себя программное обеспечение, аппаратные средства и платформы, которые служат основой для разработки, развертывания и эксплуатации сложных информационных систем. Эти инструменты используются для решения задач обработки данных, машинного обучения, управления инфраструктурой, разработки приложений и обеспечения информационной безопасности.

Одним из самых значимых инструментов в вычислительных технологиях является облачные платформы, такие как Amazon Web Services (AWS), Microsoft Azure и Google Cloud Platform (GCP). Эти платформы предоставляют доступ к вычислительным ресурсам по запросу, что позволяет компаниям и разработчикам избегать капитальных затрат на создание собственной инфраструктуры. Облако предоставляет широкий спектр сервисов, начиная от базового хранения данных и виртуальных машин до высокоуровневых сервисов, таких как базы данных, аналитические инструменты и машинное обучение. Например, сервисы AWS, такие как Amazon S3 для хранения данных или Amazon EC2 для запуска виртуальных серверов, позволяют масштабировать инфраструктуру в зависимости от потребностей бизнеса, что особенно важно в условиях переменной нагрузки.

Другим важным инструментом являются платформы для машинного обучения и искусственного интеллекта. Такие фреймворки, как TensorFlow, PyTorch и Scikit-learn, предоставляют инструменты для разработки и тренировки моделей машинного обучения, которые могут быть применены в широком спектре приложений — от компьютерного зрения до обработки естественного языка. Эти инструменты позволяют автоматизировать процессы, которые ранее требовали человеческого вмешательства, такие как классификация изображений или прогнозирование на основе больших данных. Например, TensorFlow, разработанный Google, активно используется в системах рекомендаций, обработке изображений и речевых интерфейсов. Он предоставляет высокоуровневые API для работы с нейронными сетями, а также интеграцию с облачными сервисами, что позволяет масштабировать вычисления для работы с большими наборами данных.

Также широко применяются технологии контейнеризации, такие как Docker и Kubernetes, которые революционизировали управление и развертывание приложений. Контейнеры позволяют изолировать приложения и их зависимости в стандартизированных пакетах, что делает развертывание в разных средах более предсказуемым и стабильным.

Docker предоставляет возможность создавать контейнеры, которые можно легко переносить между средами разработки, тестирования и эксплуатации, что значительно упрощает цикл разработки программного обеспечения. Kubernetes, в свою очередь, предоставляет инструменты для оркестрации контейнеров в масштабах кластера, автоматизируя развертывание, масштабирование и управление приложениями. Это делает его ключевым компонентом при построении распределённых систем и микросервисных архитектур, которые активно используются в современных приложениях.

Для управления данными и их анализа применяются инструменты, такие как Apache Hadoop и Apache Spark. Эти технологии предназначены для работы с большими данными и предоставляют платформы для распределённой обработки и анализа огромных объемов информации. Hadoop, например, предоставляет хранилище HDFS (Hadoop Distributed File System) и платформу для выполнения распределённых вычислений с помощью MapReduce, что позволяет обрабатывать терабайты данных параллельно на большом количестве серверов. Apache Spark, с другой стороны, предлагает более современный подход к распределённым вычислениям, поддерживая оперативную обработку данных в памяти, что значительно ускоряет вычисления. Эти инструменты находят применение в таких областях, как анализ больших данных, прогнозная аналитика и обучение моделей машинного обучения на масштабных данных.

Важным аспектом разработки и эксплуатации приложений является системы управления базами данных (СУБД). Реляционные базы данных, такие как MySQL, PostgreSQL и Microsoft SQL Server, остаются основными инструментами для хранения структурированных данных, обеспечивая высокую надёжность, консистентность и возможность выполнения сложных SQL-запросов. Однако, с ростом объема неструктурированных данных, всё более популярными становятся NoSQL базы данных, такие как MongoDB, Cassandra и Redis, которые предоставляют более гибкие модели хранения и масштабирования. Например, MongoDB использует документ-ориентированную модель, которая лучше подходит для хранения неструктурированных данных, таких как JSON-документы, что особенно полезно в приложениях, работающих с динамическими данными.

Для разработки программного обеспечения используются интегрированные среды разработки (IDE), такие как Visual Studio, PyCharm, Eclipse и IntelliJ IDEA. Эти среды обеспечивают разработчиков всеми необходимыми инструментами для написания, отладки и тестирования программного обеспечения. Они поддерживают интеграцию с системами контроля версий, такими как Git, что позволяет эффективно управлять исходным кодом и совместной разработкой в командах. Средства автоматизированного тестирования и CI/CD

(continuous integration and continuous delivery) системы, такие как Jenkins, Travis CI и GitLab CI, играют ключевую роль в современных подходах к разработке, обеспечивая автоматическое тестирование и развертывание приложений, что значительно снижает риск ошибок и ускоряет выпуск новых версий программного обеспечения.

Не менее важными инструментами в области информационной безопасности являются системы шифрования и защиты данных. Алгоритмы, такие как AES (Advanced Encryption Standard) и RSA, обеспечивают защиту данных на уровне шифрования, гарантируя их конфиденциальность и целостность при передаче и хранении. Также используются инструменты для мониторинга и предотвращения атак, такие как Snort, Wireshark и Splunk, которые помогают отслеживать подозрительную активность в сети и защищать критически важные системы от внешних угроз [5].

Сравнение теории и практики в области вычислительных технологий является важным аспектом для понимания того, как фундаментальные теоретические концепции применяются в реальных системах и какие ограничения и вызовы возникают при их внедрении. Теория информатики предоставляет абстрактные модели и методы для разработки алгоритмов, структур данных и вычислительных процессов, в то время как практическое использование этих методов часто сталкивается с ограничениями, вызванными реальными условиями, такими как ограниченные ресурсы, нестабильные данные и сложность реализации.

Одним из ключевых аспектов теоретической информатики является анализ сложности алгоритмов. Теория предлагает множество методов для оценки производительности алгоритмов с использованием асимптотической сложности, позволяя понять, как алгоритмы ведут себя при увеличении объема входных данных. Например, алгоритмы сортировки могут иметь различную временную сложность: сортировка вставками работает за $O(n^2)$, тогда как сортировка слиянием — за $O(n \log n)$. Однако на практике выбор алгоритма часто зависит не только от его асимптотической сложности, но и от таких факторов, как доступная память, реализация на конкретной архитектуре и характер входных данных. Например, хотя сортировка слиянием асимптотически более эффективна, для небольших массивов данных сортировка вставками может работать быстрее из-за меньших накладных расходов на управление памятью.

Параллельные вычисления — ещё одна область, где теория и практика часто различаются. С теоретической точки зрения параллельные алгоритмы могут значительно ускорить выполнение задач, разделяя вычисления на несколько процессоров [2]. Например, алгоритмы на основе модели PRAM (Parallel Random Access Machine) предполагают, что процессоры могут одновременно обращаться к общей памяти без задержек, что в теории даёт

значительное ускорение. Однако в реальных системах накладные расходы на синхронизацию и передачу данных между процессорами могут существенно снизить ожидаемую производительность. В практике параллельных вычислений важно учитывать такие факторы, как задержки при доступе к памяти, ограничение полосы пропускания сети и необходимость синхронизации задач. Это приводит к тому, что на практике параллельные алгоритмы часто работают не так эффективно, как предсказывается теоретическими моделями.

Теория сложности предоставляет важные классы задач, такие как P, NP, NP-полные задачи, которые классифицируют вычислительные задачи по их решаемости за полиномиальное время. Например, задачи класса NP-полных, такие как задача коммивояжера, в теории не могут быть решены за полиномиальное время, если $P \neq NP$. На практике, однако, существуют эвристические и приближённые алгоритмы, которые позволяют находить решения для конкретных случаев этих задач с приемлемой точностью за разумное время. Например, в задачах логистики и маршрутизации активно используются методы, такие как алгоритмы ближайшего соседа или генетические алгоритмы, которые, хотя и не гарантируют оптимального решения, тем не менее предоставляют достаточно качественные результаты для применения на практике. Это пример того, как на практике обходят теоретические ограничения с помощью методов, адаптированных к конкретным задачам.

Машинное обучение также демонстрирует значительные различия между теорией и практикой. Теоретически, обучающие модели, такие как нейронные сети или поддерживающие векторы, описываются через строгие математические модели, которые определяют их способность к обобщению данных. Однако на практике разработка и обучение таких моделей требует значительных вычислительных ресурсов и большого количества данных, что может существенно ограничивать возможности их применения [4]. Например, обучение глубоких нейронных сетей требует высокопроизводительных графических процессоров и дней или недель вычислений, что ограничивает их использование в реальном времени или в малых компаниях с ограниченными ресурсами. Кроме того, на практике часто возникает проблема переобучения, когда модель слишком точно подстраивается под обучающие данные и плохо работает на новых, что не всегда очевидно на этапе теоретического проектирования модели.

Вычислительная геометрия предлагает множество эффективных алгоритмов для решения геометрических задач, таких как триангуляция, выпуклые оболочки и пересечение отрезков. Эти алгоритмы часто имеют хорошие теоретические оценки по времени выполнения, такие как алгоритм Грэхема для построения выпуклой оболочки с временной сложностью $O(n \log n)$. Однако на практике возникают дополнительные сложности при работе

с реальными данными, такими как точность вычислений, численные ошибки и некорректные входные данные. Например, при работе с географическими информационными системами (ГИС) ошибки округления могут приводить к тому, что точка, которая теоретически должна принадлежать объекту, оказывается вне его границ. Это требует дополнительных методов обработки ошибок и проверки точности, что усложняет практическую реализацию теоретических алгоритмов.

Заключение

Теория баз данных и системы управления базами данных (СУБД) также демонстрируют различия между теорией и практикой. В теории реляционные базы данных, основанные на математической теории множеств, предоставляют средства для хранения и манипулирования данными через SQL-запросы, обеспечивая строгое соблюдение нормализаций данных и поддержание целостности. Однако на практике реляционные базы данных могут испытывать проблемы с масштабированием при работе с большими объемами данных, что привело к развитию нереляционных (NoSQL) баз данных. Такие системы, как MongoDB и Cassandra, предлагают более гибкие схемы данных и горизонтальное масштабирование, что позволяет эффективно работать с большими объемами неструктурированных данных. В данном случае теория реляционных баз данных, хотя и остается фундаментальной, на практике дополняется новыми подходами для удовлетворения современных требований к производительности и гибкости.

Список литературы

1. Гаврилов Г.П. Алгоритмы и структуры данных: учебное пособие. Москва: Изд-во МГУ, 2010. 312 с.
2. Колесников В.И., Шабунин П.А. Параллельные вычисления: теория и практика. Санкт-Петербург: Питер, 2017. 456 с.
3. Суслов С.С., Михайлов И.В. Искусственный интеллект: модели и методы. Москва: Физматлит, 2018. 408 с.
4. Гудфеллоу И., Бенгио Ю., Курвиль А. Глубокое обучение. Кембридж, Массачусетс: MIT Press, 2016. 775 с.
5. Таненбаум А.С., Уэзералл Д. Компьютерные сети. 5-е изд. Аппер-Седл-Ривер, Нью-Джерси: Pearson Education, 2011. 960 с.

THE MAIN THEORETICAL AND PRACTICAL APPROACHES IN COMPUTER SCIENCE AND COMPUTING TECHNOLOGIES USED IN THE DEVELOPMENT OF SOFTWARE AND INFORMATION SYSTEMS

Borisov Artyom Andreevich

Student of the Far Eastern Federal University
Vladivostok, Russian Federation

Rostotsky Maxim Viktorovich

Student of the Far Eastern Federal University
Vladivostok, Russian Federation

Shorokhov Konstantin Dmitrievich

Student of the Far Eastern Federal University
Vladivostok, Russian Federation

Kuchuk Maxim Igorevich

Student of the Far Eastern Federal University
Vladivostok, Russian Federation

Gardash Vladislav Viktorovich

Student of the Far Eastern Federal University
Vladivostok, Russian Federation

Abstract. This article examines the differences between theoretical and practical approaches in the field of computing technologies. The object of research is the information economy. The subject of the study is the system elements of the information economy. The purpose of the study is to reveal the main theoretical and practical approaches in computer science and computing technologies used in the development of software and information systems. Theoretical models such as algorithm complexity analysis, parallel computing theory, and complexity theory provide the fundamental principles for developing effective solutions. However, in practice, these theories face limitations caused by real-world conditions such as resource availability, data complexity, and performance requirements. The article analyzes key examples such as the application of parallel algorithms, machine learning, computational geometry and databases, with an emphasis on how theoretical concepts adapt to practical tasks. As a result of the research, strategies are identified to bridge the gap between theory and practice, and the importance of hybrid approaches in modern computer science is emphasized.

Key words: theoretical computer science, computing technologies, parallel computing, machine learning, algorithms, innovations, new technologies.

References

1. Gavrilov G.P. Algorithms and data structures: a textbook. Moscow: Publishing House of Moscow State University, 2010. 312 p.
2. Kolesnikov V.I., Shabunin P.A. Parallel computing: theory and practice. St. Petersburg: Peter, 2017. 456 p.
3. Suslov S.S., Mikhailov I.V. Artificial intelligence: models and methods. Moscow: Fizmatlit, 2018. 408 p.
4. Goodfellow I., Bango Y., Courville A. Deep learning. Cambridge, Massachusetts: MIT Press, 2016. 775 p.
5. Tanenbaum A.S., Weatherall D. Computer networks. 5th ed. Upper Saddle River, New Jersey: Pearson Education, 2011. 960 p.