На правах рукописи

Егунов Виталий Алексеевич

МОДЕЛИ И МЕТОДЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ ДЛЯ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ В ГЕТЕРОГЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ

2.3.7. Компьютерное моделирование и автоматизация проектирования

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

доктора технических наук

Волгоград – 2025

Работа выполнена на кафедре «Системы автоматизированного проектирования и поискового конструирования» в федеральном государственном бюджетном образовательном учреждении высшего образования «Волгоградский государственный технический университет».

Научный консультант доктор технических наук, профессор, **Кравец Алла Григорьевна.**

Официальные оппоненты: **Заборовский Владимир Сергеевич,** доктор технических наук, профессор, ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого», Высшая школа технологий искусственного интеллекта, профессор;

Топорков Виктор Васильевич, доктор технических наук, профессор, ФГБОУ ВО «Национальный исследовательский университет «МЭИ», кафедра «Вычислительные технологии», заведующий;

Кондусова Валентина Борисовна, доктор технических наук, доцент, ФГБОУ ВО «Оренбургский государственный университет», кафедра «Системы автоматизации производства», доцент.

Ведущая организация **ФГБУН "Санкт-Петербургский Федеральный исследовательский центр Российской академии наук".**

Защита состоится «___» _____ 2025 г. в $\_\_^{00}$ часов на заседании диссертационного совета 24.2.282.08, созданного на базе ФГБОУ ВО «Волгоградский государственный технический университет» по адресу: 400005, г. Волгоград, пр. Ленина, д. 28, ауд. 209.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВО «Волгоградский государственный технический университет» и на официальном сайте https://www.vstu.ru по ссылке https://www.vstu.ru/upload/iblock/399/399314bbff80a9bbe163510c9fa7dd64.pdf.

Автореферат разослан « ___ » _____ 2025 г.

Учёный секретарь
диссертационного совета Асанова Наталия Васильевна

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

В настоящее время проблеме разработки эффективных программных систем (ЭПС), используемых в гетерогенных вычислительных средах (ГВС) с различной архитектурой для решения вычислительно-сложных задач в научных и инженерных расчетах, системах искусственного интеллекта, уделяется большое внимание. ГВС представляет собой вычислительную среду, содержащую несколько вычислительных ресурсов, причем множества операций, выполняемых различными ресурсами, различны, каждый ресурс имеет свою функциональную специализацию. С точки зрения проектирования программной системы под гетерогенностью понимается программирование различных устройств, имеющих собственные системы команд. Программная система (ПС) - сложный технический объект, представляющий собой совокупность взаимодействующих программных компонентов. ПС является частью вычислительной среды (ВС), представляющей собой сочетание программной и аппаратной частей, при этом под повышением эффективности ПС понимается минимизация затрат, связанных с ПС на различных этапах жизненного цикла (ЖЦ), включая затраты на разработку, модификацию и эксплуатацию ПС. Эффективность ПС тесно связана с понятием эффективности использования ПС вычислительных ресурсов, которая определяется степенью загрузки вычислительных ресурсов (компонентов ВС), участвующих в процессе вычислений. Эффективность получаемых ПС и ряд других характеристик в значительной степени зависят от квалификации разработчика. На начальных этапах проектирования разработчики часто используют эвристический подход к повышению эффективности ПС, для улучшения характеристик полученных ПС используются различные инструментальные средства. Оба указанных подхода не гарантируют получения необходимого результата, кроме того, указанные методы проектирования могут привести к увеличению времени разработки и последующей модификации ПС.

Необходимо отметить, что современная российская действительность ориентирована на отечественные решения, вырос спрос на импортозамещение, в т.ч. в сегменте высокопроизводительных вычислений (ВПВ) для решения инженерных задач, к которым относятся в т.ч. вычислительные ядра для численного моделирования, статистического анализа, машинного обучения, геометрические ядра CAD/CAE-систем. Кроме того, особое значение теме исследования придают прогнозы, которые предрекают значительный рост доли ВПВ на рынке информационных технологий в ближайшие годы и даже десятилетия.

В 2024 году Указом Президента РФ утверждена Стратегия научно-технологического развития Российской Федерации. Цель стратегии — «обеспечить устойчивое, динамичное и сбалансированное развитие России на долгосрочный период». Одним из направлений стратегии, позволяющих получить значимые научные и научно-технические результаты, создать отечественные наукоемкие технологии является «переход к передовым технологиям проектирования и создания высокотехнологичной продукции, основанным на

применении интеллектуальных производственных решений, роботизированных и высокопроизводительных вычислительных систем, новых материалов и химических соединений, результатов обработки больших объемов данных, технологий машинного обучения и искусственного интеллекта».

Правительство РФ в 2020 г. утвердило Программу фундаментальных научных исследований в Российской Федерации на долгосрочный период (2021 - 2030 годы), в которой подчеркивается необходимость «создания методов, алгоритмов, инструментальных средств и пакетов прикладных программ для ВС сверхвысокой производительности», разработки «теоретических основ вычислительных методов и алгоритмов для компьютерных систем высокой производительности». Подчеркивается важность применения ВПВ, которые окажут большое влияние на развитие фундаментальных наук.

Таким образом, тематика исследования является актуальной.

Степень разработанности темы исследования.

В процессе разработки программных средств в соответствии с ГОСТ Р ИСО/МЭК 12207-2010 различают два типа работ (видов деятельности): системные и программные. При выполнении работ программного типа широко используются средства автоматизации проектирования ПС, представляющие собой набор методов и инструментальных средств, известный как Computer-Aided Software Engineering (CASE) (ГОСТ Р ИСО/МЭК 14764-2002). В состав CASE входят средства анализа, проектирования и программирования ПС, CASE-средства определяются как программные средства для поддержки жизненного цикла ПС. Технологии CASE призваны повышать эффективность процессов проектирования и модификации, способствовать повышению надёжности и упрощению сопровождения созданных при помощи CASE-технологий ПС.

Разработке методов, предназначенных для повышения эффективности ПС для ГВС с различной архитектурой, а также реализация этих методов в виде CASE-технологий посвящены работы ряда исследователей. Так, например, коллективами под руководством Воеводина В.В,, Воеводина Вл.В., Гергеля В.П., Каляева И.А., Котенко И.В., Саенко И.Б., Самарского А.А., Топоркова В.В. и др. проведено множество исследований по распараллеливанию вычислений, разработке методов и алгоритмов параллельной обработки данных, исследований в области высокопроизводительных вычислений. Большое количество исследований посвящено использованию специализированных ускорителей, ВС с реконфигурируемой архитектурой (работы Каляева И.А., Заборовского В.С., Левина И.И., Ронжина А.Л., Якубовского М.В., и др.) Широкое внимание исследователей привлекают проблемы исследования и повышения эффективности алгоритмов при решении конкретных прикладных задач. Данные вопросы освещены в работах Воеводина В.В,, Воеводина Вл.В,, Гергеля В.П., Ортега Дж., Деммеля Дж., Голуба Дж., Ван Лоуна Ч., Донгарра Дж., Глинского Б.М., и др.

В то же время стоит отметить гораздо меньшее число исследований по кеш-оптимизации и векторизации вычислений, разработке моделей, предназначенных для повышения эффективности проектируемых ПС, и созданию на основе этих моделей CASE-средств. Методы, используемые для повышения эффективности

векторизации вычислений, освещены в работах: С.Ларсен, Дж.Лиу, (Superword Level Parallelism, SLP), Д.Нузман, И.Розен, А.Закс (векторизация обработки чередующихся данных с промежутками), А. Эйхенбергер, П.Ву, К.О'Брайен (исследование вопросов выравнивания данных), Й.Чен, Ч.Мендис и др.(использование Single Instruction Multiple Operation Multiple Data, SIMOMD), Д.Нузман, А.Закс, С.Багсорхи, (построение оптимизирующих компиляторов с блоками автовекторизации, Intel C++ Compiler, GCC) и др. В области разработки моделей проектирования ПС и методов, основанных на этих моделях известны исследования, связанные с разработкой моделей ВС с иерархической структурой памяти (С.Уильямс, Дж.Трейбиг, Х.Сентегель, К.Йотов, Т.Лоу, М.Фриго и др.). Методы, используемые для улучшения стратегии кеширования данных описываются в работах: Ф.Густавсон, Дж.Маккомб (множественная реализация алгоритмов и эмпирический поиск параметров), Р.Уэйли, Дж.Доггараа, Д.Билмс, Дж.Деммель (автонастройка с использованием генерации кода), Д.Спампинато, Г.Белтер, К.Ванг (автонастройка с использованием аналитических техник), К.Йотов, Т.Лоу (аналитический поиск параметров), и др. Абсолютное большинство предложенных моделей направлено на решение задачи умножения плотных матриц. При этом аналитические подходы, описанные в ряде работ, не предполагают автоматизации, для другого преобразования необходимо проводить похожее исследование. Отмечается также сложность проведения подобного анализа, требующая наличия высокой квалификации и ряда компетенций у исследователя-разработчика. Для большинства подходов необходим непосредственный доступ к ВС, на которой планируется использование исследуемых ПС. Кроме того, предложенные методы получения необходимых характеристик проектируемых алгоритмов предполагают аналитическое моделирование, при этом моделирование часто проводится для конкретного семейства микропроцессоров, авторы методик не предложили информационных систем, автоматизирующих данный процесс.

Таким образом, сложилась ***проблемная ситуация***, сущность которой заключается в том, что:

- отсутствуют или являются неполными существующие подходы к разработке ЭПС, которые могут использоваться на начальном этапе проектирования и верификации алгоритмов
- исследования и разработки в сфере автоматизации проектирования ПС часто имеют частный характер;
- подходы к повышению эффективности ПС, основанные на кеш-оптимизации и векторизации вычислений, являются неполными и имеют существенные несовершенства
- существует необходимость разработки отечественных программных решений, направленных на проектирование ЭПС.

В проектировании ПС для ГВС выявлена **фундаментальная проблема**: научно-технические подходы к проектированию ЭПС обладают существенной неполнотой и имеют частный характер, что оказывает существенное влияние на выбор стратегии повышения эффективности ПС.

Научно-техническая проблема исследования заключается в разработке нового методологического подхода к автоматизации проектирования ПС, связанного с созданием моделей, методов и алгоритмов, предназначенных для повышения эффективности ПС для решения инженерных задач в ГВС на различных этапах ЖЦ.

Цели и задачи.

Целью работы является разработка научных основ поддержки жизненного цикла проектируемых программных систем для решения инженерных задач в гетерогенных вычислительных средах.

Для достижения цели исследования в работе поставлены и решены следующие **задачи:**

1. Исследовать существующие научно-технические подходы, применяемые для проектирования и повышения эффективности ПС.
2. Разработать методологию проектирования ПС для решения инженерных задач в ГВС, повышающую эффективность ПС на всех этапах ЖЦ, включающую основные проектные процедуры, модели и методы:
   - компьютерные модели оптимального проектирования ПС для ГВС;
   - методы повышения эффективности ПС для ГВС.
3. Верифицировать созданную методологию проектирования ЭПС на примере практически значимых вычислительно-сложных задач.
4. Разработать архитектурные и программные решения системы поддержки жизненного цикла проектируемых ПС.

В работе в качестве **объекта исследования** выступает жизненный цикл ПС для решения инженерных задач в ГВС, а **предметом исследования** являются модели, методы и алгоритмы автоматизации проектирования для повышения эффективности ПС в ГВС на различных этапах жизненного цикла ПС.

Научная новизна. Впервые предложена совокупность моделей и методов поддержки жизненного цикла программных систем для решения инженерных задач в гетерогенных вычислительных средах, реализующая комплексный подход к повышению эффективности программных систем, включающая в себя:

1. Модель улучшения показателей эффективности программных систем, основанная на методах статического анализа исходного кода и оценки аппаратурно-временной сложности ПС, отличающаяся от известных переносом этапа анализа эффективности ПС с динамического анализа готового приложения на статический анализ исходного кода, позволяющая снизить количество итераций проектных процедур на этапах разработки и модификации жизненного цикла ПС (п. 3, 10 паспорта специальности).
2. Математическая модель оценки эффективности проектируемых программных систем, основанная на классификации подсистем целевой вычислительной архитектуры, отличающаяся от известных совокупностью новых метрик ПС и обоснованием их применимости, что позволяет на различных этапах ЖЦ комплексно оценить ПС при эксплуатации в гетерогенных вычислительных средах. (п. 6 паспорта специальности).

3. Аналитический метод определения максимального размера блока при регулярном доступе к данным, основанный на отображении множества кешируемых данных на множества кеш-памяти, отличающийся от известных прямым способом расчета параметров регулярных шаблонов доступа к данным, позволяющий выбирать параметры алгоритмов, минимизирующие количество кеш-промахов при доступе к данным проектируемой ПС. (п. 6 паспорта специальности).

4. Имитационная модель исполнения программы в гетерогенных вычислительных средах с иерархической структурой памяти, основанная на множествах системных и пользовательских операций, где впервые использован полный набор параметров, характеризующих различные уровни подсистемы памяти, что позволяет определять характеристики программных систем на конкретной целевой вычислительной архитектуре, а также характеристики подсистемы памяти для проектируемой ПС. (п. 8 паспорта специальности).

5. Комплекс методов повышения эффективности программных систем, основанный на предложенных моделях (п. 6 паспорта специальности)::

– метод использования подсистемы памяти, в котором впервые реализован аналитический подход к оценке эффективности проектируемых ПС на базе совокупности метрик ПС, позволяющий осуществить улучшение стратегии кеширования данных в гетерогенных вычислительных средах с иерархической структурой памяти;

– метод векторизации вычислений, в котором впервые реализован новый способ векторной обработки данных, что позволяет избежать потерь производительности программных систем в гетерогенных вычислительных средах при доступе к данным;

– метод использования GPU, в котором впервые реализован аналитический подход к оценке эффективности проектируемых программных систем на базе совокупности метрик ПС, позволяющий повысить уровень загрузки и степень сбалансированности загрузки потоковых процессоров.

6. Методология проектирования эффективных программных систем для решения инженерных задач в гетерогенных вычислительных средах, основанная на исследовании жизненного цикла ПС, впервые реализующая подход непрерывного улучшения показателей проектируемых ПС, отличающаяся от известных реализацией предложенных моделей и методов, которая позволяет принимать проектные решения по эффективному использованию вычислительных ресурсов на начальных этапах жизненного цикла ПС (п. 1,4 паспорта специальности).

Теоретическая и практическая значимость работы.

Теоретическая значимость диссертационного исследования заключается в развитии методологических подходов к поддержке жизненного цикла программных систем с высокой степенью локальности данных, а именно разработана и верифицирована методология проектирования ЭПС для решения инженерных задач в гетерогенных вычислительных средах. Методология

позволяет повысить эффективность объекта проектирования, сократить время и автоматизировать проектирование ПС для вычислительно-сложных процедур.

Практическая значимость диссертационного исследования подтверждается применением разработанной методологии проектирования при создании ЭПС**.** Полученные результаты позволяют решать задачи автоматизации проектирования и повышения эффективности ПС для ГВС в области ВПВ инженерного и научного характера, включая ПС для систем искусственного интеллекта. На основе предложенных моделей разработаны архитектурные и программные решения, Предложена архитектура системы, содержащая программную реализацию статического анализатора кода программы, использующего методы частично-динамического анализа, позволяющая снизить затраты на проектирование ПС, одновременно снижая требования к уровню компетенций разработчика. Реализованы новые методы улучшения стратегии кеширования данных в ГВС с иерархической структурой памяти, новые алгоритмы повышения эффективности векторизации вычислений, использования GPU. Предложенные методы и алгоритмы существенно повышают эффективность проектируемых ПС, включая их энергоэффективность.

Разработанная методология проектирования ЭПС для решения инженерных задач в ГВС и программные решения позволяют:

– осуществлять статический анализ исходного кода с целью непрерывного улучшения показателей эффективности получаемых ПС;

– производить оценку критериев эффективности получаемых ПС, а также параметров алгоритмов на начальных стадиях их проектирования и верификации с использованием разработанных метрик ПС;

- получать прирост производительности и снижение энергопотребления проектируемых ПС до 4 раз за счет улучшения стратегии кеширования данных и эффективных алгоритмов векторизации вычислений.

Методология и методы исследования. Для решения поставленных задач в работе используются методы объектно-ориентированного программирования, вычислительной математики и теории алгоритмов, методы математического и имитационного моделирования, методы автоматизированного проектирования и CASE-технологий.

Положения, выносимые на защиту.

1. Модель улучшения показателей эффективности программных систем, позволяющая снизить количество итераций проектных процедур на этапах разработки и модификации ЖЦ ПС.

2. Математическая модель оценки эффективности проектируемых ПС, позволяющая на различных этапах ЖЦ комплексно оценить ПС при эксплуатации в гетерогенных вычислительных средах.

3. Аналитический метод определения максимального размера блока при регулярном доступе к данным, позволяющий производить исследование показателей эффективности использования подсистемы памяти проектируемых ПС.

4. Имитационная модель исполнения программы в ГВС с иерархической структурой памяти, позволяющая определять технические характеристики ПС на конкретной целевой вычислительной архитектуре, а также характеристики подсистемы памяти для проектируемой ПС.

5. Комплекс методов повышения эффективности ПС:

- метод повышения эффективности использования подсистемы памяти, позволяющий принимать проектные решения по улучшению стратегии кеширования данных в ГВС с иерархической структурой памяти;
- метод повышения эффективности векторизации вычислений, позволяющий избежать потерь производительности ПС при доступе к данным;
- метод повышения эффективности использования GPU, позволяющий повысить уровень загрузки и степень сбалансированности загрузки потоковых процессоров.

6. Методология проектирования ЭПС для решения инженерных задач в ГВС, позволяющая на начальных этапах ЖЦ ПС принимать проектные решения по эффективному использованию вычислительных ресурсов.

7. Концепция и архитектура системы поддержки жизненного цикла проектируемых ПС на основе предложенных моделей и методов, которая позволяет обеспечить повышение эффективности ПС.

Степень достоверности и апробация результатов.

Достоверность результатов диссертации обеспечивается корректным использованием методов исследований и математического аппарата, а также подтверждается результатами проведенных экспериментов и успешного внедрения разработанных ПС. Реализуемые модели, методы и алгоритмы строго аргументированы и критически оценены по сравнению с другими известными результатами.

Диссертационная работа выполнена в рамках одного из научных направлений Волгоградского государственного технического университета, а также проектов РНФ № 25-21-20073 – «Модели и методы проектирования эффективного программного обеспечения для высокопроизводительных вычислений» (руководитель), РФФИ № 18-47-340010 - "Разработка, адаптация и анализ эффективности алгоритмов реализации базовых операций линейной алгебры на неоднородных многопроцессорных вычислительных архитектурах с потоковыми FPGA – ускорителями" (руководитель), № 16-47-340385 – «Адаптация алгоритмического обеспечения ядра системы междисциплинарного моделирования с многотельным представлением технических объектов для неоднородных многопроцессорных вычислительных архитектур» (исполнитель),

Диссертационная работа выполнена в рамках ряда проектов, выполненных по заказу ЗАО «Топ Системы» (№ 44/227-19 – «Разработка специализированных объектно-ориентированных библиотек классов, предназначенных для решения геометрических задач прототипа CAD системы, связанных с генерацией, модификацией и анализом трёхмерных математических моделей»); ООО «ТЕСИС» (№ 44/776-15 – «Модификация алгоритмов реализации матричных операций с использованием векторных расширений систем команд, дополненных

совмещенными сложениями–умножениями (FMA3), и архитектур процессоров с длинным командным словом»); ООО "ОКБ АНТ", ОАО «Волжский абразивный завод» и др., в т.ч. ряда проектов, связанных с разработкой ПС по теме диссертации, в которых автор являлся научным руководителем или исполнителем. Получены 3 акта о внедрении результатов исследования.

Основные положения диссертационной работы докладывались и обсуждались на квалификационной секции международной научной конференции «Математические методы в технике и технологиях» ММТТ-37 (2024, Казань), международных конференциях ММТТ (2025, Самара, 2024, Казань, 2023; Нижний Новгород, 2022; Ярославль), «Суперкомпьютерные дни в России (Russian Supercomputing Days)» (2016, 2019, Москва), «Параллельные вычислительные технологии (ПаВТ)» (2021, Волгоград; 2020, Пермь; 2016, Архангельск), «Национальный Суперкомпьютерный Форум (НСКФ)» (2015-2019, Переславль-Залесский), «Creativity in Intelligent Technologies and Data Science (CIT&DS)» (2015 – 2023, Волгоград), «International Scientific Conference Artificial intelligence and digital technologies in technical systems (AIDTTS)» (2020, Волгоград), «Перспективные информационные технологии (Advanced Information Technologies and Scientific Computing)» (2020, Самара), «International Conference on Artificial Life and Robotics (ICAROB)» (2018, Япония), «Инновационные, информационные и коммуникационные технологии (ИНФО)» (2024, Сочи, 2023, Махачкала; 2022, Сочи).

Материалы диссертационного исследования легли в основу курсов «Программная инженерия», «Параллельные вычислительные технологии», «Применение ускорителей и оптимизация приложений в системах искусственного интеллекта».

Публикации.

Основные результаты диссертации опубликованы в 170 научных работах, в том числе 2 монографиях, 45 – в изданиях, рекомендованных ВАК России, 20 публикаций в международных изданиях, проиндексированных в Scopus и Web of Science и 3 отчетах о НИР. Получены 7 свидетельств о регистрации программ на ЭВМ.

Структура и объем работы.

Диссертация состоит из введения, пяти глав, заключения, библиографического списка из 195 наименований и 2 приложений. Общий объем диссертации 414 страниц, основная часть содержит 119 рисунков и 40 таблиц.

Соответствие паспорту научной специальности.

Полученные в диссертационной работе результаты соответствуют следующим пунктам паспорта специальности 2.3.7. Компьютерное моделирование и автоматизация проектирования, а именно пункту 1 - методология компьютерного моделирования и автоматизированного проектирования в технике и технологиях, включая постановку, формализацию и типизацию проектных и технологических процедур, алгоритмов и процессов проектирования; пункту 3 - разработка научных основ построения комплекса средств САПР, включающего информационное, математическое,

лингвистическое, методическое, техническое, программное обеспечение непрерывной информационной поддержки жизненного цикла проектируемых объектов пункту; 4 - разработка принципиально новых и повышение эффективности существующих методов и средств взаимодействия проектировщик – система, включая компьютерные модели и технологии искусственного интеллекта; пункту 6 - разработка компьютерных моделей, алгоритмов, программных комплексов оптимального проектирования технических изделий и процессов; пункту 8 - разработка имитационных компьютерных моделей для оценки и тестирования технических, экономических, экологических характеристик технических объектов проектирования; пункту 10 - разработка научных основ реализации жизненного цикла проектирование – производство – эксплуатация – утилизация, построения интегрированных средств управления проектными работами и унификации прикладных протоколов информационной поддержки.

Сведения о личном вкладе автора.

Личный вклад автора заключается в разработке основных теоретических положений, выносимых на защиту; в разработке моделей, методов и методологии, вошедших в структуру концепции; в разработке концепции и архитектуры системы поддержки жизненного цикла ПС. Вклад автора в основные опубликованные работы был определяющим. Все представленные в диссертации положения, выносимые на защиту, получены лично автором.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследований, сформулированы цель и задачи работы, научная новизна, теоретическая и практическая значимость полученных результатов.

В первой главе выполнен анализ научно-технических подходов к проектированию ЭПС. Рассмотрены модели жизненного цикла ПС, основные методологии проектирования ПС, CASE-технологии как средства автоматизации проектирования ПС. Исследованы вопросы, связанные с поддержкой жизненного цикла ПС в области ВПВ, влиянием методологий проектирования на качество получаемых ПС.

Под ВПВ понимают применение ВС высокой производительности для решения вычислительно-сложных задач. Технологии ВПВ в последнее время реализуются с помощью вычислительных кластеров, представляющих собой группу компьютеров, объединенных каналами связи. Компьютеры кластера совместно решают вычислительно-сложные задачи, поэтому с точки зрения пользователя представляют собой единый аппаратный ресурс.

ГОСТ Р ИСО/МЭК 12207-2010 устанавливает связь между системой в целом и программным средством. В соответствии с ГОСТ Р ИСО/МЭК ТО 15271-2002 система является «комбинацией технических средств, компьютеров, программных средств, материалов, персонала и возможностей». Стандарт описывает, как система становится комбинацией технических и программных средств. Это означает, что при создании работоспособной системы учитываются как аппаратные, так и программные компоненты.

ВС, включая вычислительные кластеры, являясь системой, представляет собой комбинацию аппаратных и программных средств. Для автономных систем аппаратные средства представляют собой совокупность вычислительных ресурсов ВС. Для вычислительных кластеров под данным термином понимается совокупность отдельных компьютеров, узлов кластера, и высокоскоростных каналов связи, объединяющих отдельные узлы в единую вычислительную среду. Программные средства вычислительного кластера включают в себя:

- специализированные ПС, представляющие собой инструменты и системы, используемые для реализации и создания высокопроизводительных ВС;
- ПС, проектируемые для решения вычислительно–сложных задач.

Для достижения требуемой производительности в процессе проектирования ПС для вычислительных кластеров необходимо решить несколько технических задач:

- максимально эффективно использовать вычислительные ресурсы отдельных узлов вычислительного кластера для повышения эффективности проектируемых ПС в рамках отдельных узлов;
- распределить вычисления между узлами вычислительного кластера за счет распараллеливания вычислений.

Диссертационное исследование направлено на разработку методологии проектирования ПС, которые обрабатывают большие массивы данных и осуществляют регулярный доступ к данным в соответствии с определенными шаблонами. К подобным ПС относятся:

- ПС для решения научных и инженерных задач, требующих большого времени вычислений, включая вычислительные ядра для численного моделирования, статистического анализа, машинного обучения, обработки больших данных и т.д..
- ПС для решения инженерных задач, требующие минимального времени отклика ВС, включая CAD/CAE-системы, а также реализацию геометрических ядер, ПС обработки мультимедиа-контента и т.д.

Данные ПС обладают высокой степенью локальности данных и проектируются для использования как в автономных ВС (компьютерах), так и в вычислительных кластерах в зависимости от назначения и условий эксплуатации.

Объектом проектирования являются программные системы для решения инженерных задач, повышающие эффективность использования вычислительных ресурсов ВС, в т.ч. узлов вычислительного кластера. Данные ПС относятся к прикладным ПС, разрабатываются на компилируемых языках программирования, относятся к проблемно-ориентированным ПС, предназначенным для решения научных и инженерных задач.

Эффективность использования вычислительных ресурсов определяется степенью загрузки вычислительных ресурсов (компонентов ВС), участвующих в процессе вычислений.

$$E = \frac{O_R * K}{O_T} \quad (1)$$

где E – эффективность использования вычислительного ресурса, O_T – общее количество операций, формирующих множество $S_O = \{O_T\}$, выполняемых в процессе

исполнения ПС, потенциально использующих вычислительный ресурс, O_R – количество операций из множества S_O, использующих данный ресурс, формируют множество $S_R = \{O_R\}$, K - коэффициент, принимающий значения от 0 до 1 и характеризующий способ выполнения операций из множества S_R.

Средневзвешенный параметр эффективности можно определить следующим образом (2)

$$E_{av} = \frac{\sum_{i=1}^{N}(k_i * E_i)}{N} \quad (2)$$

где k_i - весовые коэффициенты, E_i – эффективность *i-ой* подсистемы, N - количество подсистем.

В соответствии с ГОСТ 23501.108-85 установлен ряд признаков классификации систем автоматизированного проектирования (САПР), включая тип объекта проектирования, где одной из классификационных группировок является САПР программных изделий. Характеристика данной классификационной группировки в соответствии со стандартом: «Проектирует программы для электронных вычислительных машин, станков с ЧПУ, роботов и т.д.».

На рисунке 1 показаны компоненты системы, используемые в процессе автоматизации проектирования.

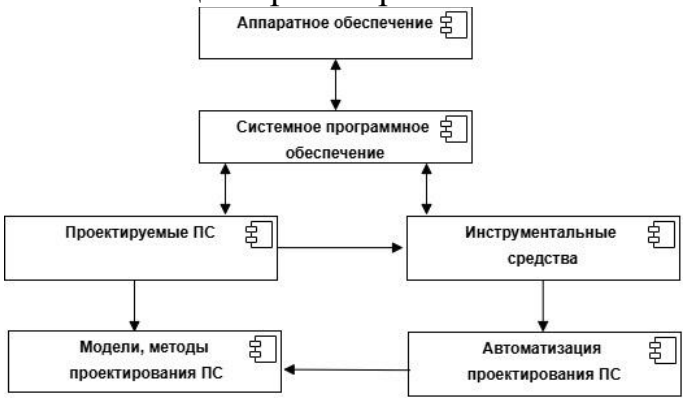


Рисунок 1 - Автоматизация проектирования ПС

Проектируемые ПС взаимодействуют с системными ПС и проектируются с использованием инструментальных средств. Для автоматизации проектирования необходимо разработать средства автоматизации проектирования. В свою очередь для решения этой задачи необходимо разработать новые модели и методы проектирования ПС.

Исследованы методы анализа ПС, выделены две группы подобных методов: статический и динамический анализ ПС. Показано, что потенциально динамический анализ позволяет выявить большее количество ошибок в программе, проанализировать ее эффективность, однако, само проведение анализа относится к собранной и отлаженной ПС, в который заложены возможные неэффективные проектные решения. Избежать выполнения неудачных итераций в процессе проектирования ПС можно, если включить анализ эффективности ПС в ранние этапы проектирования, например, в проведение статического анализа исходного кода на этапе разработки ПС

Проанализированы критерии эффективности ПС и влияние архитектуры ВС на эффективность ПС, Рассмотрены группы критериев эффективности, зависящих от типа ПС, критерии эффективности, характеризующие ПС как технический объект, подлежащий разработке и модификации. Приведены основные понятия и определения предметной области. Определено понятие ПС как технического объекта. Определена целевая вычислительная архитектура, в качестве которой

рассматривается ГВС, где под ГВС понимается ВС, содержащая несколько вычислительных ресурсов, причем множества операций, выполняемых различными ресурсами, различны. Методология проектирования ЭПС реализована в ГВС, имеющей в своем составе следующие вычислительные ресурсы: вычислительная система с общей памятью, построенной по иерархической схеме (CPU), векторный процессор или векторное расширение команд (SIMD), графический процессор (GPU). Каждый из ресурсов характеризуется набором параметров.

Проведен анализ влияния архитектуры ВС на эффективность ПС, в т.ч. анализ влияния иерархической структуры памяти, распараллеливания вычислений, реализации распределенной модели вычислений, векторизации вычислений, использования ускорителей (GPU и FPGA). Отмечено, что на этапе разработки ПС в рамках ЖЦ ПС при проектирования архитектуры ПС особое внимание уделяется разработке и модификации моделей: моделей, используемых, в разработке, управляемой моделями (Model-driven Development, MDD), моделей процесса разработки ПС (RUP, Scrum), моделей ЖЦ проектирования ПС (Software Development Lifecycle Models, SDLC), На том же этапе ЖЦ ПС при написании программного кода одними из ключевых упоминаемых аспектов являются «техники программирования», где под этим термином исследователи упоминают различные способы оптимизации кода, используемые для повышения эффективности ПС. В качестве критериев эффективности ПС на этапе проектирования архитектуры наибольшую частоту упоминаний имеет «Производительность», «Повторное использование» и «Ремонтопригодность», т.е. сложность модификации ПС. На этапе реализации наибольшую частоту упоминаний имеет «Производительность» и «Эффективность». Среди языков программирования наиболее используемыми в области ВПВ являются C/C++ и Fortran.

Сформулированы выводы о необходимости разработки новых подходов к проектированию ЭПС для решения инженерных задач.

Во второй главе проведен анализ современных методов, а также реализации данных методов в виде CASE-технологий повышения эффективности ПС для решения инженерных задач. Выполнен обзор моделей в области ВПВ, используемых для повышения эффективности ПС, обзор методов, используемых для улучшения стратегии кеширования данных, векторизации вычислений, использования GPU, а также инструментальных средств, применяемых для анализа и улучшения характеристик ПС.

Основное внимание уделяется моделям исполнения, которые исследуют поведение ВС. Рассматриваются модели ВС с иерархической структурой памяти, акцент делается на том, что некоторые подобные подходы основаны на моделировании производительности целевой ВС. Рассматриваются модели двух категорий: моделирование “черного ящика”, которое опирается на статистические методы и машинное обучение для описания и прогнозирования производительности ВС на основе наблюдаемых данных и моделирование “белого ящика”, которое использует упрощенные математические модели для

описания взаимодействия кода с оборудованием. К недостаткам большинства подобных моделей можно отнести привязку к конкретному МП и сложность проведения аналитического моделирования, необходимость проведения нового исследования для каждого нового исследуемого алгоритма.

Среди методов улучшения стратегии кеширования данных выделены использование пространственной и временной локальности данных, в т.ч. оптимизация доступа и расположения данных в памяти, блочная реализация алгоритмов. Проанализированы параметризованные модели алгоритмов, в т.ч. эмпирический поиск параметров, автоматическая настройка с генерацией кода, использование аналитических моделей. Исследованы вопросы применения кеш-симуляторов. Исследованы вопросы автоматической векторизации кода современными оптимизирующими компиляторами, ограничения блоков автоматической векторизации. Проанализированы методы автоматической векторизации циклов, векторизации зависимостей и чередующихся данных с промежутками, вопросы влияния выравнивания данных на эффективность векторизации кода, созданные на основе этих методов CASE-средства.

Рассмотрены инструментальные средства, которые используются для анализа и улучшения характеристик ПС для ГВС, в т.ч. оптимизирующие компиляторы, высокопроизводительные библиотеки, инструменты статического и динамического анализа.

Сформулированы выводы о необходимости разработки комплекса средств поддержки ЖЦ ПС для решения инженерных задач в ГВС.

В третьей главе проводится разработка основных положений методологии проектирования ПС для решения инженерных задач в ГВС, повышающей эффективность ПС на всех этапах жизненного цикла. Под повышением эффективности ПС понимается минимизация затрат, связанных с ПС на различных этапах ЖЦ, включая затраты (например, затраты на оборудование, разработчиков и т.д.) на разработку, модификацию и эксплуатацию ПС.

Проведено исследование ЖЦ ПС в аспекте создания и эксплуатации ПС в области ВПВ для решения инженерных и научных задач, для которых важными являются вопросы выбора алгоритма решения задачи и повышения эффективности ПС.

ЖЦ ПС в соответствии с ГОСТ Р ИСО/МЭК 12207 определяется как развитие системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии разработки концепции и заканчивая прекращением применения. Для ПС жизненный цикл определяется как период времени, который начинается с момента принятия решения о необходимости создания ПС и заканчивается в момент ее полного изъятия из эксплуатации. Стандарт определят понятие процесса как совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы. Стандарт группирует различные виды деятельности, которые могут выполняться в течение жизненного цикла программных систем, в семь групп процессов. На рисунке 2 приведены процессы ЖЦ, анализируемые в диссертационной работе и их соответствие категориям процессов ЖЦ в соответствии со стандартом:

- процесс разработки включает в себя все процессы ЖЦ, связанные с проектированием ПС, включая процессы проекта, технические процессы и процессы реализации ПС;

- процесс эксплуатации включает в себя процесс функционирования ПС (технические процессы) и процессы верификации и валидации ПС (процессы поддержки);

- процесс модификации включает в себя процесс решения проблем в ПС (процессы поддержки).

ЖЦ проектируемых ПС состоит из трех этапов: разработки, модификации и эксплуатации (рисунок 3).

Рисунок 2 - Процессы ЖЦ, анализируемые в диссертационной работе и их соответствие категориям процессов ЖЦ в соответствии со стандартом

После окончания этапа разработки ПС переходит на этап эксплуатации. В процессе эксплуатации ПС может возникнуть необходимость в модификации ПС по причинам недостаточной эффективности ПС, необходимости адаптации ПС под новые вычислительные архитектуры, необходимость устранения выявленных ошибок и т.д. Эффективность ПС на каждом этапе ЖЦ при этом характеризуется кортежем (3).

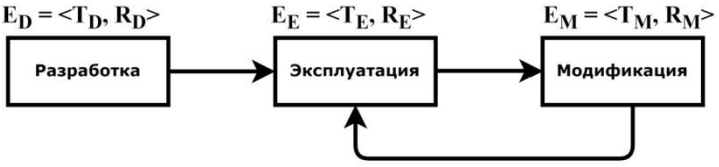


Рисунок 3 – Жизненный цикл ПС

$$E = \langle T, R \rangle \quad (3)$$

где E – эффективность, T – затрачиваемое время, R – затрачиваемые ресурсы. Для этапов разработки и модификации время T представляет собой время, затрачиваемое командой разработчиков на выполнение всех работ, необходимых для завершения текущего этапа, R – затрачиваемые ресурсы, в т.ч. затраты на разработчиков и затраты на оборудование. На этапе эксплуатации T – время работы ПС для решения конкретной задачи, R – используемые при этом ресурсы, в т.ч. состав аппаратных средств, потребляемые энергоресурсы.

Так как снизить время на этапе эксплуатации ЖЦ можно путем увеличения количества и характеристик используемых ресурсов, в качестве количественной оценки степени эффективности ПС будем использовать меру аппаратурно-временной сложности алгоритма (4).

$$C = O(Q) * O(T) \quad (4)$$

где C – мера аппаратурно-временной сложности, $O(Q)$ – мера аппаратурной сложности реализации алгоритма, $O(T)$ – мера временной сложности реализации алгоритма.

Таким образом, повышение эффективности ПС на этапе эксплуатации можно представить в виде задачи снижения аппаратурно-временной сложности

программной реализации решаемой задачи. Потребляемые ВС энергоресурсы в процессе решения задачи напрямую зависят от значения ее аппаратурно-временной сложности. Аналогичным образом можно оценить эффективность ПС на этапах разработки и модификации. В данном случае показатель ресурсов является комплексным и зависит не только от используемых аппаратных средств, но и от числа и квалификации занятых работников.

Предлагаемую модель улучшения показателей эффективности ПС можно описать следующим образом (5).

$$E = \langle T, R \rangle$$
$$E = f(a), a = (a_1, a_2, \dots, a_n) \in A \quad (5)$$
$$E_{i+1} \geq E_i$$

Эффективность ПС на этапах разработки и модификации зависит от времени и ресурсов, затраченных на реализацию этапа, в то же время эффективность разработки зависит от программной реализации решаемой задачи из множества А программных реализаций. Модель описывает улучшение показателей эффективности проектируемой ПС, при этом эффективность ПС на итерации разработки *i+1* будет не хуже эффективности на итерации *i*.

На рисунке 4 представлен этап разработки ПС в соответствии с моделью улучшения показателей эффективности ПС, реализованный в виде итеративного процесса, итерации которого представлены в двух вариантах: традиционного (AS-IS) и предлагаемого (TO-BE).

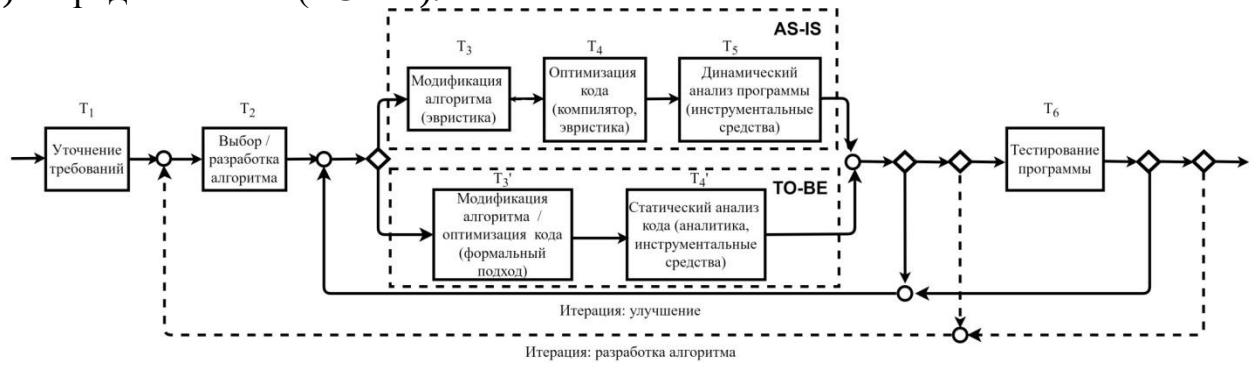


Рисунок 4 - Этап разработки ПС в соответствии с моделью улучшения показателей эффективности ПС

Отличие представленных подходов к реализации этапа разработки ПС заключается в переносе этапа анализа эффективности ПС с динамического анализа готового приложения на статический анализ исходного кода, что позволяет снизить количество итераций на этапах разработки и модификации жизненного цикла программных систем (таблица 1).

Из таблицы 1 видно, что в связи с уменьшением общего числа итераций и уменьшения совокупного времени реализации отдельных проектных процедур этапа разработки, предлагаемый подход оказывается эффективнее традиционного.

На рисунке 5 представлен этап модификации ПС в соответствии с моделью улучшения показателей эффективности ПС. На этапе модификации ПС предлагаемый подход также оказывается эффективнее традиционного в связи с уменьшением общего числа итераций и уменьшения совокупного времени реализации отдельных проектных процедур.

Таблица 1 – Сравнение традиционного и предлагаемого подходов к реализации этапа разработки ПС.

Традиционный подход	Предлагаемый подход
$T = T_1 + \sum_{i=1}^{n}\left(a_i * T_2 + \sum_{j=1}^{m_i}\left[b_{i,j} * T_2 + (T_3 + T_4 + T_5)\right] + T_6\right)$ T_1- время на уточнение требований T_2 - время на выбор / разработку алгоритма T_3 - время на модификацию алгоритма T_4 - время на оптимизацию кода T_5 - время на динамический анализ программы T_6 - время на тестирование программы $a_i \in \{0,\ 1\}, b_{i,j} \in \{0,\ 1\}, a_1 = 1,$ $b_{i,\,1} = 0,$ если $a_i = 1$	$T = T_1 + \sum_{i=1}^{n'}\left(a'_i * T_2 + \sum_{j=1}^{m'_i}\left[b'_{i,\,j} * T_2 + (T'_3 + T'_4)\right] + T_6\right)$ T'_3- время на модификацию алгоритма / оптимизацию кода T'_4- время на статический анализ кода $a'_i \in \{0,\ 1\}, b'_{i,\,j} \in \{0,\ 1\}, a'_1 = 1,$ $b'_{i,\,1} = 0,$ если $a'_i = 1$
$T'_3 + T'_4 < T_3 + T_4 + T_5$ $n' \le n, m' \le m$	

Кроме того, на данном этапе выделен отдельный предлагаемый вариант модификации ПС, основанных на параметризованных алгоритмах. Статический анализ кода в данном случае обеспечивает подбор нужных параметров за одну итерацию, что также эффективнее традиционного подхода с динамическим анализом готового приложения.

Рисунок 5 - Этап модификации ПС в соответствии с моделью улучшения показателей эффективности ПС

На рисунках 4 и 5 не приведены очевидные действия, выполняемые в процессе разработки ПС, такие как предварительный статический анализ исходного кода, компиляция кода, поиск и устранения ошибок, т.к. данные действия являются обязательными и напрямую не оказывают влияния на итоговую эффективность проектируемой ПС.

В традиционном подходе эффективность ПС исследуется в процессе динамического анализа после получения работоспособной версии ПС, при этом анализируется уже готовый код, в который заложены возможные неэффективные проектные решения. Статический анализ традиционно используется на начальных

этапах для поиска ошибок, дефектов и уязвимостей ПС. В предлагаемом подходе предлагается включить анализ эффективности ПС в ранние этапы проектирования, в частности, в этап разработки исходного кода.

Предлагаемый подход позволяет повысить эффективность ПС на этапе разработки за счет уменьшения общего времени реализации этапа. Это становится возможным за счет улучшения показателей эффективности в результате применения формальных подходов к модификации алгоритма и оптимизации кода, что в целом приводит к сохранению или уменьшению количества итераций разработки.

Для принятия проектных решений по повышению эффективности проектируемых ПС в рамках модели улучшения показателей эффективности была предложена математическая модель оценки эффективности проектируемых ПС в ГВС с различными характеристиками. В рамках данной модели предложены метрики, позволяющие осуществлять оценку эффективности ПС на различных этапах ЖЦ (рисунок 6).

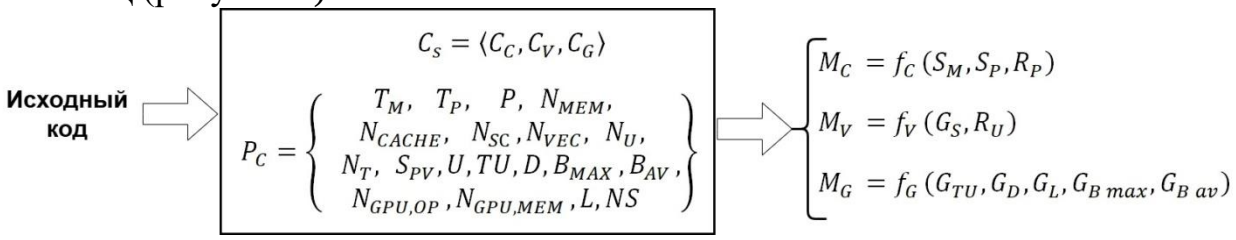


Рисунок 6 - Математическая модель оценки эффективности проектируемых ПС,

где C_S – характеристики ГВС, в т.ч. C_C – характеристики подсистемы памяти, C_V – характеристики подсистемы векторизации вычислений, C_G – характеристики GPU; P_C – , множество определяемых параметров ПС, в т.ч. T_M – время работы с памятью, T_P – время выполнения программы, P – усредненное значение вероятности кеш-промаха, N_{MEM} - количество обращений к основной памяти, N_{CACHE} - количество обращений к кеш-памяти, N_{SC} – необходимое число скалярных операций, N_{VEC} – число операций с учетом векторизации, N_U – число невыровненных векторных обращений к памяти, N_T– общее число векторных обращений к памяти, S_{PV} – потенциальное ускорение, U – доля невыровненных операций, TU – доля использованных потоков, D – средняя плотность вычислений, B_{max} – максимальный уровень несбалансированности потоков, B_{av} – средний уровень несбалансированности потоков, $N_{GPU,OP}$ - число операций в потоках GPU, $N_{GPU,MEM}$ – число операций обращения к памяти в потоках GPU, L – степень локальности данных, NS – число потоковых обращений к памяти/

На вход модели подается исходный код проектируемой ПС, для которого определяются параметры, необходимые для вычисления соответствующих метрик ПС. Возможно формирование трех групп метрик:

- M_C – группа метрик, характеризующая эффективность стратегии кеширования данных;
- M_V – группа метрик, характеризующая эффективность векторизации вычислений;
- M_G – группа метрик, характеризующая эффективность использования GPU.

Способ оценки отдельных метрик ПС приведен в таблице 2.

Таблица 2 – Метрики математической модели оценки эффективности ПС в ГВС с различными характеристиками

Метрика	Способ оценки
Ускорение программы	$S_P = {}^{T_{P1}}/_{T_{P2}}, T_P = \sum_{s=1}^{N_s}\left(N_{s,op} * t_{op} + \sum_{g=1}^{NG_s} N_{s,g}\left(\alpha_{s,g} * t_{mem} + +\left(1-\alpha_{s,g}\right) * t_{cache}\right)\right)$
Ускорение работы с памятью	$S_M = {}^{T_{M1}}/_{T_{M2}}, T_M = \sum_{s=1}^{N_s}\sum_{g=1}^{NG_s} N_{s,g}\left(\alpha_{s,g} * t_{mem} + \left(1-\alpha_{s,g}\right) ** t_{cache}\right)$
Снижение вероятности кеш-промаха	$R_P = {}^{P_1}/_{P_2}, P = {}^{N_{mem}}/_{(N_{mem}+N_{cache})}$ $N_{mem} = \sum_{s=1}^{N_s}\sum_{g=1}^{NG_s}\left(N_{s,g} * \alpha_{s,g}\right),\ N_{cache} = \sum_{s=1}^{N_s}\sum_{g=1}^{NG_s}\left(N_{s,g} * \left(1-\alpha_{s,g}\right)\right)$
Увеличение потенциального ускорения	$G_S = {}^{S_{PV2}}/_{S_{PV1}}, S_{PV} = {}^{N_{SC}}/_{N_{VEC}},$
Снижение доли невыровненных обращений	$R_U = {}^{U_1}/_{U_2}, U = {}^{N_U}/_{N_T},$
Увеличение доли использованных потоков	$G_{TU} = {}^{TU_2}/_{TU_1}.\ TU = \sum_{i=1}^{NB} NT_i \Big/ (NB * NT_b)$
Увеличение средней плотности вычислений	$G_D = {}^{D_2}/_{D_1}, D = \left(\sum_{b=1}^{NB}\sum_{t=1}^{NT_b} N_{op,b,t}/N_{mem,b,t}\right)\Big/(NB * NT_b)$
Увеличение средней степени локальности данных	$G_L = {}^{L_2}/_{L_1}, L = \left(\sum_{b=1}^{NB}\sum_{w=1}^{NW_b} NS_{b,w}/N_{b,w}\right)\Big/(NB * NW_b)$
Увеличение степени сбалансированности потоков	$G_{B,max} = \frac{B_{1,max}}{B_{2,max}}, B_{max} = \max_w\left(\max_i\left(N_{op,i,w}\Big/\sum_{k=1}^{NT_w} N_{op,k,w}\right)\right)$ $G_{B,av} = \frac{B_{1,av}}{B_{2,av}}, B_{av} = \sum_{w=1}^{NW}\sum_{i=1}^{NT_w}\left(N_{op,i,w}\Big/\sum_{k=1}^{NT_w} N_{op,k,w}\right)\Big/(NW * NT_w)$

где N_s – число шагов алгоритма, s – номер шага алгоритма, $N_{s,op}$ – число операций обработки данных, выполняемых на s-м шаге алгоритма;, t_{op}– среднее время выполнения операций обработки данных, NG_s – число подмножеств, содержащих, операции обращения к памяти с равными значениями вероятности кеш-промаха в рамках s-го шага алгоритма, g – номер подмножества, содержащего операции обращения к памяти в рамках s-го шага алгоритма, $N_{s,g}$– число операций в подмножестве номер g, $\alpha_{s,g}$ – вероятность кеш-промаха при обращении к памяти в подмножестве g при выполнении s-го шага алгоритма, t_{mem} – значение времени обращения к основной памяти на целевой вычислительной архитектуре, t_{cache} – значение времени обращения к кеш-памяти на целевой вычислительной архитектуре (ускорение программы), NB - число блоков потоков, NT_b – число потоков в блоке, NT_i– число использованных потоков в i-м блоке (увеличение доли использованных потоков), $N_{op,b,t}$– число операций обработки данных в t-м потоке b-го блока, $N_{mem,b,t}$– число операций обращения к

памяти в t-м потоке b-го блока (увеличение средней плотности вычислений), NW_b – число варпов в блоке, $NS_{b,w}$– число потоковых обращений к памяти в w-м варпе b-го блока, $N_{b,w}$– общее число обращений к памяти в w-м варпе b-го блока (увеличение средней степени локальности данных), $N_{op,i,w}$ - число операций в i-м потоке w-го варпа, NT_w– число потоков в варпе, NW – общее число варпов (увеличение степени несбалансированности потоков).

В третьей главе в качестве примера приведен прикладной протокол определения метрик ПС для оценки эффективности использования подсистемы памяти для различных паттернов циклового доступа к элементам массива, результат представлен в таблице 3.

В процессе получения метрик в таблице 3 учитывалось влияние аппаратной предвыборки данных (6)

$$N'_{mem} = N_{mem} * \big(b + g * (1 - b)\big), \quad (6)$$

где N'_{mem} – количество обращений к памяти, оцененное с учетом предвыборки данных, b – доля неуспешных предвыборок, g – доля времени на считывание предвыбранной ранее строки от полного времени обращения к памяти без предвыборки.

Таблица 3 – Усредненное значение вероятности кеш-промаха при цикловом доступе к массиву

Паттерн доступа	Вероятность кеш-промаха
Паттерн 1 – ijk; обращение вдоль строк матрицы, N проходов вдоль каждой строки	$\frac{(1-k)}{N} + \frac{k}{N^3}$
Паттерн 2 - jki; обращение вдоль столбцов матрицы, N проходов вдоль матрицы по столбцам	$\frac{(1-l)*c}{N} + \frac{l*\mathrm{c}}{8*N}$
Паттерн 3 - ikj; один проход вдоль матрицы по строкам, в рамках каждого прохода N обращений к каждому элементу	$\frac{1}{8*N}$
Паттерн 4 - jik; обращение вдоль строк матрицы, N проходов вдоль матрицы	$\frac{1}{N^2}$
Паттерн 5 - kji; обращение вдоль столбцов, N проходов вдоль каждого столбца	$\frac{(1-l)*c}{N} + \frac{l*\mathrm{c}}{8*N}$
Паттерн 6 - kij; один проход вдоль матрицы по столбцам, в рамках каждого прохода N обращений к каждому элементу	$\frac{(1-l)}{N} + \frac{L}{8*N}$

где $N * N$ – размерность массива, k – параметр, учитывающий размер строки массива, l – параметр, учитывающий размер столбца массива, c – параметр, учитывающий снижение эффективности столбцовых паттернов доступа в множественно-ассоциативную кеш-память.

Рассчитанные параметры базовых паттернов доступа были использованы для оценки эффективности программных реализаций задач BLAS (Basic Linear Algebra Subroutines) 3 уровня. Проведенные вычислительные эксперименты полностью подтверждают выводы, сделанные на основе рассчитанных значений метрик.

Была исследована проблема оценки значений вероятности «холодных» или обязательных кеш-промахов. В случае непрерывного расположения данных (рисунок 7) вероятность кеш-промаха определяется в соответствии в (7) и (8).

Важным в данном случае является вопрос определения максимального размера блока, при котором генерируются только «холодные» промахи и не происходит вытеснения строк из кеш-памятим, т.к. данный вопрос исследуется в процессе разработки блочных алгоритмов.

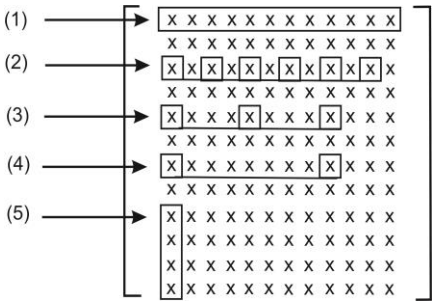


Рисунок 7 – «Холодные» промахи

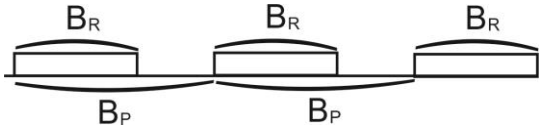
$$P = \frac{1}{(L/B)} \quad (7)$$

$$P = \frac{1}{k} \quad (8)$$

где P – вероятность кеш-промаха, B – длина (sizeof) переменной используемого типа данных, L – длина кеш-линии, k – число элементов, используемых в кеш – линии.

Выражение (7) используется для непрерывно расположенных данных (вариант 1 на рисунке 7) , выражение (8) для считываний с интервалами (варианты 2-5 на рисунке 7).

В вариантах 1-4 (рисунок 7) кеш-память заполняется равномерно, следовательно, максимальный размер такого блока совпадает с размером кеш-памяти. Однако, в случае регулярного доступа к данным, которые не расположены в памяти непрерывно (рисунок 8), это не так, потому что кеш-память заполняется неравномерно. Для оценки размера блока B_B в данном случае был предложен аналитический метод определения максимального размера блока при регулярном доступе к данным (9). Алгоритм реализации выражения (9) представлен на рисунке 9. и соответствует применению метода в общем случае, в т.ч. для сложной структуры шаблона или нерегулярного доступа в память.



$$B_B: \mathrm{C}_B \rightarrow \mathrm{C}_M, \quad \forall c \leq M, \mathrm{c} \in \mathrm{C}_M \quad (9)$$

Рисунок 8– Регулярный доступ к памяти

где M – степень ассоциативности кеш-памяти, B_P – длина паттерна доступа, B_R – считываемая часть паттерна доступа, C_B - множество кешируемых элементов B_P, C_L - множество кеш-линий, C_M– множества кеш-памяти.

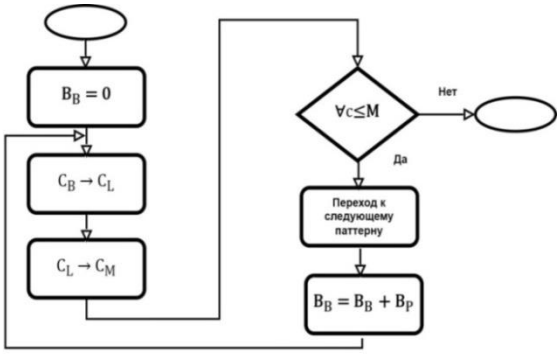


Рисунок 9 – Метод количественной оценки метрик ПС

1. Текущая длина блока равна 0.
2. Отобразить множество кешируемых элементов текущего паттерна на множество кеш-линий.
3. Отобразить множество кеш-линий на множества кеш-памяти
4. Если число заполненных элементов во всех множествах кеш-памяти не больше степени ассоциативности, перейти к п.5, иначе к п.8.
5. Переход к следующему паттерну
6. Текущую длину блока увеличить на длину паттерна
7. Перейти к п.2
8. Конец

В случае регулярной структуры шаблона, размер которого не превосходит размера кеш-памяти, все количественные характеристики метрик ПС можно получить аналитическим способом, включая и максимальную длину блока данных. Для реализации аналитического применения данного метода были

сформулированы и доказаны четыре теоремы, сформулированы следствия из теорем, необходимые для определения отображения множеств C_B и C_M.

Теорема о формировании рядов взаимно простых чисел (теорема 1). Если x и y – положительные взаимно простые числа, то числовой ряд, полученный как

$$L = (a * x) \,\%\, y, \quad a = \overline{0, y-1}, \tag{10}$$

будет содержать значения

$$L = \underline{\overline{0, y-1}}, \tag{11}$$

и не содержать повторов.

Доказательство. Используется доказательство от противного. Допустим, утверждение теоремы неверно, ряд L содержит другие значения. Значения элементов ряда также лежит в пределах от 0 до (y-1), что очевидно (элементы формируются как остаток от деления на y), однако ряд содержит повторяющиеся значения. Тогда можно подобрать значения a, b, c, m, n, такие, что верно следующее

$$a * x = m * y + c, \tag{12}$$
$$b * x = n * y + c,$$

и два элемента ряда будут равны c, т.к. появляются одинаковые значения остатков от деления (a*x)%y и (b*x)%y для разных a и b. Пусть a>b, вычтем из первого уравнения второе, получим

$$(a - b) * x = (m - n) * y. \tag{13}$$

В левой и правой части выражения (13) – общие кратные x и y. Но, т.к. они являются взаимно простыми числами, то их наименьшее общее кратное (НОК) равно x*y. Однако, по условию a<y, b<y, оба коэффициента являются положительными числами и, т.к. a>b, (a-b) является положительным числом, меньшим y. Подобрать такие значения невозможно, следовательно, сделанное предположение неверно, а верно утверждение, сделанное в теореме.

Следствие 1 из теоремы 1. Если x и y – положительные взаимно простые числа, то выводы теоремы 1 справедливы также для числового ряда

$$L = (a * x + b) \,\%\, y, \quad a = \overline{0, y-1},\ b = \overline{0, x-1}. \tag{14}$$

Следствие 2 из теоремы 1. Если x и y – положительные взаимно простые числа, то числовой ряд, полученный использованием (10) без ограничений на значения a, будет содержать конкатенацию рядов (11). Другими словами, в полученном ряду элементы, отстоящие друг от друга на y позиций, будут содержать одни и те же значения.

Следствие 3 из теоремы 1. Максимальная длина ряда, сформированного с использованием (10) без ограничений на значения a и не содержащего повторов, равна y.

Теорема о заполняемости упорядоченных множеств, мощности которых являются взаимно простыми числами (теорема 2). Пусть L и M – положительные взаимно простые числа, есть последовательность чисел (список) A, формируемая как конкатенация последовательностей (списков) A’ длины L со следующей структурой

$$A' = [0, \dots, L-1]. \tag{15}$$

Пусть также есть список B’ длины L, содержащий одну единицу в позиции k, остальные нули. Пусть есть упорядоченное множество B мощности M, первоначально содержащее нулевые значения. Пусть осуществляется обход

списка A, для каждого элемента вычисляется функция f, модифицирующая элементы B.

$$f: B(i\%M) = B(i\%M) + B'(i\%L)\,, \qquad (16)$$

где i – номер элемента A. Тогда существует конечное число S пройденных элементов A, после которых все элементы B будут содержать единицы.

Доказательство. Пусть длина списка A равна НОК(L, M), т.е. S=L*M в данном случае, тогда каждый элемент B модифицируется в соответствии с (16) L раз. В самом деле, в соответствии со следствием 3 из теоремы 1, максимальная длина ряда в списке A, сформированного с использованием (10) без ограничений на значения a и не содержащего повторов, равна L. Значение i в выражении (16) принимает значения

$$i = \overline{0, L * M - 1}\,, \qquad (17)$$

При этом список индексов B в (16) представляет собой конкатенацию числовых рядов (11) для разных значений b. составленных следующим образом

$$i_B = (a * L + b)\,\%M\,, \quad a = \overline{0, M-1}, \quad b = \overline{0, L-1}., \qquad (18)$$

Здесь a – номер строки длиной M внутри числового ряда, b – номер элемента внутри строки, i_B– индекс внутри строки. Каждая такая строка в соответствии со следствием 3 из теоремы 1 содержит все индексы B и не содержит повторов. Число строк равно L, следовательно, каждый элемент B модифицируется L раз.

В случае, если длина списка A равна S=L*M, к каждому элементу B в процессе модификации прибавляются значения всех элементов B’ по одному разу. Если принять в выражении (14) x=M, y=L, b – номер элемента B, получится числовой ряд длины L, не содержащий повторов, представляющий собой набор всех индексов в B’. Таким образом, к каждому элементу B (с номером b в (14)) в процессе модификации прибавляются все значения B’ по одному разу. Так как только один элемент B’ с индексом k содержит единицу, а остальные содержат нули, то после обхода L*M элементов списка A все элементы B будут содержать единичные значения.

Следствие из теоремы 2. Если в условии к теореме 2 длина списка равна S=L*M, список B’ содержит единицы в v позициях, в остальных содержит нули, v<=L, тогда после прохода A все элементы B будут содержать v.

Теорема о формировании рядов чисел, не являющихся взаимно-простыми (теорема 3). Представляет собой обобщение теоремы 1 на случаи, когда x и y не являются взаимно простыми числами.

Теорема о заполнении числами упорядоченных множеств, мощности которых не являющимися взаимно-простыми (Теорема 4). Пусть L и M не являются взаимно простыми числами, тогда сформировать равномерно заполненное одинаковыми значениями множество B в соответствии с условием теоремы 2 невозможно.

Интерпретация параметров, используемых в теоремах: L – длина блоков (шаг в памяти в элементах кеш-линий), M – число множеств в множественно-ассоциативном кеше, B’ – указывает на то, какие элементы в шаблонах кешируются, B – множества кеша; значения элементов соответствуют количеству заполненных банков в конкретном множестве, A - память. Тогда с учетом блок-

схемы (рисунок 9), теорем (1-4) и следствий из них определить максимальный размер блока, не вызывающий кеш-промахов, можно следующим образом (рисунок 10).

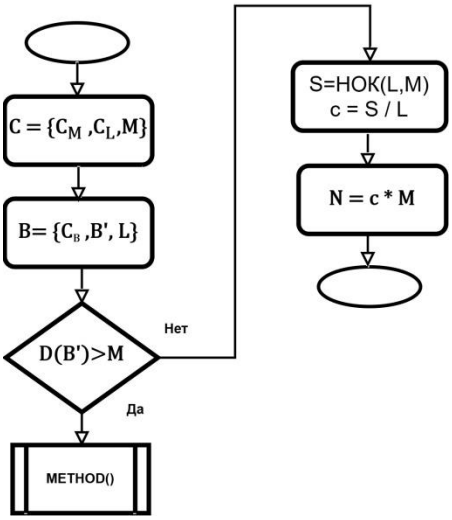


Рисунок 10 – Метод определения максимального размера блока, не вызывающего кеш-промахов

1. Определить характеристики анализируемого уровня кеш-памяти (в частности, определить количество множеств в каждом банке M).
2. Определить характеристики шаблона доступа к памяти (длину шаблона в кеш-линиях L, количество кеш-линий в шаблоне, к которым производится обращение – количество единиц в множестве B').
3. Если количество заполненных элементов в множестве B' превышает степень ассоциативности анализируемого уровня кеш-памяти, перейти к п.7
4. Определить S=HOK(L, M);
5. С учетом степени ассоциативности определить максимальный размер блока N.
6. Перейти к п.8
7. Реализовать итерационный вариант метода (рисунок 9)
8. Конец.

Была осуществлена экспериментальная проверка метода количественной оценки метрик ПС, подтвердившая его достоверность. Вычислительные эксперименты проводились на узлах вычислительного кластера ВолгГТУ, который представляет собой гетерогенный вычислительный кластер, узлы которого являются неоднородными, при этом в качестве базовых узлов для проведения экспериментов использовались ВС на базе микропроцессоров Xeon E5-2650 v3(x2) 2.3 GHz, оборудованные GPU RTX 3060 и Nvidia P100.

Предложена имитационная модель исполнения программы (ИМИП) в ГВС с иерархической структурой памяти, позволяющая определять характеристики ПС на заданной целевой вычислительной архитектуре, а также конфигурацию подсистемы памяти для проектируемой ПС и параметров алгоритмов (рисунок 11).

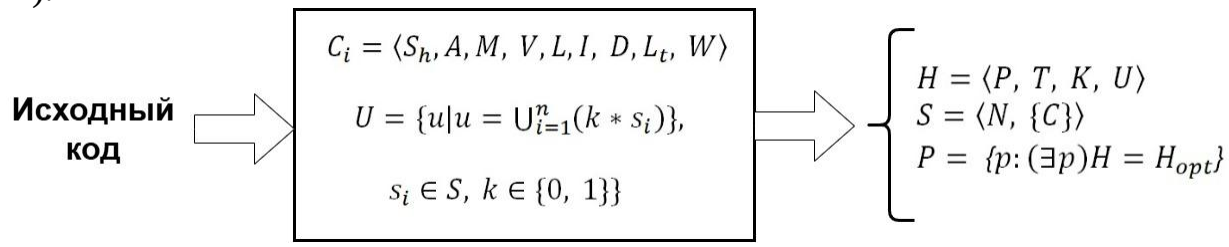


Рисунок 11 - Имитационная модель исполнения программы в ГВС с иерархической структурой памяти

ИМИП представляет собой детерминированную модель, описывающую ГВС с иерархической структурой памяти, содержащей N уровней, каждый уровень реализован в виде кеш-памяти, которая описывается кортежем (19).

$$C_i = \langle S_h, A, M, V, L, I, D, L_t, W \rangle, 1 \le i \le N \quad (19)$$

где S_h- разделяемость, A – ассоциативность, M – степень ассоциативности, V – размер, L - длина кеш-линии, I – инклюзивность, D - алгоритм вытеснения, L_t- латентность, W – write through, write back.

В ИМИП определены операции двух типов:

- множество пользовательских операций U (20);
- множество системных (атомарных) операций S (таблица 4).

$$U = \{u | u = \cup_{i=1}^{n}(k * s_i)\}, s_i \in S,\ k \in \{0,\ 1\}\} \quad (20)$$

Таблица 4 - Атомарные системные операции

Обозначение	Описание
data[I][tag] = d	запись в строку кеш-памяти уровня I с тегом tag данных d;
d = data[I][tag]	считывание строки кеш-памяти уровня I с тегом tag
reg = data[1][tag]	считывание данных из L1 в регистр МП
data[1][tag] = reg	запись содержимого регистра МП в L1
d = mem[tag]	считывание строки, идентифицируемой тегом tag, из памяти
mem[tag] = d	запись строки, идентифицируемой тегом tag, в память
c = HIT(I, tag)	проверка, находится ли строка с тегом tag в уровне кеш-памяти I
tag = ALG(I)	определение тега вытесняемой строки с уровня I в соответствии с используемым алгоритмом замещения строк

На основе системных операций при имитационном моделировании формируется множество пользовательских операций. Ниже приведен псевдокод операции считывания данных из памяти.

```
read (tag, N){
  curI = 1;
  while ((! HIT(curI, tag)) && (curI <= N))   curI++;
    if  (curI <= N)    return read_cache(tag, curI, N);
    else{ return read_mem(tag, N);
}
```

При имитации исполнения программы осуществляется поиск строки в кеш-памяти, начиная с уровня L1 и ее считывание read_cache(). Если строка отсутствуют в кеш-памяти, она считывается из основной памяти. Реализация системных операций считывания из заданного уровня кеш-памяти (read_cache()) и считывания из основной памяти (read_mem()) зависит от способа организации кеш-памяти. Ниже приведен псевдокод реализации в ИМИП системных операций для кеш-памяти с инклюзивной организацией (таблица 5)..

Таблица 5 – Реализация системных операции считывания данных для кеш-памяти с инклюзивной организацией

Считывание из кеш-памяти	Считывание из памяти

read_cache(tag, I, N){
curI = 1;
tag_pushed = ALG(1);
while (curI < I){
data[curI][tag_pushed] = (data[I][tag]);
curI++;}
return data[1][tag];
}
	read_mem(tag, N){
	tag_pushed = ALG(1);
	t = mem[tag]
	curI = 1;
	while (curI < N){
	data[curI][tag_pushed] = t;
	curI++;}
	return t;
	}

На вход ИМИП подается исходный код проектируемой ПС, который преобразуется в последовательность пользовательских операций (20), выполнение

которых моделируется в ГВС с характеристиками (19). В результате формируются выходные параметры (таблица 6). ИМИП применяется в случае невозможности получения значений метрик ПС аналитическим способом (см. рисунок 10 и таблица 3).

Таблица 6 – Выходные параметры модели исполнения программы в подсистеме памяти

Наименование	Выходные параметры
Характеристики ПС на целевой вычислительной архитектуре	$H = \langle P,\ T,\ K,\ U \rangle$ P – вероятность кеш-промахов T – оценка времени выполнения K – коэффициент повторного использования данных U – процент неоптимальных обращений к памяти (штрафы)
Конфигурация подсистемы памяти для ПС	$S = \langle N,\ \{C\} \rangle$ N – число уровней кеш-памяти $\{C\}$ – параметры кеш-памяти различных уровней
Определение параметров алгоритмов	$P = \{p: (\exists p) H = H_{opt}\}$ P – множество параметров

На основе предложенных моделей был разработан комплекс методов повышения эффективности ПС:

- использования подсистемы памяти (ПП), позволяющий принимать проектные решения по улучшению стратегии кеширования данных в ГВС с иерархической структурой памяти;
- векторизации вычислений, позволяющий избежать потерь производительности ПС при доступе к данным;
- использования GPU, позволяющий повысить уровень загрузки и степень сбалансированности загрузки потоковых процессоров.

Метод повышения эффективности использования ПП (метод улучшения стратегии кеширования, МУСК) позволяет существенно повысить эффективность проектируемой ПС (рисунок 12). Метод состоит из двух этапов.

1. На первом этапе выполняются следующие действия.

1.1. Разбиение множества операций доступа к памяти U на непересекающиеся подмножества U_α с равными значениями вероятности кеш-промахов P_{cm}

1.2. Оценка числа элементов N и вероятностей кеш-промахов P в каждом подмножестве

1.3. Оценка числа операций арифметико-логической обработки данных N_{op} для всего алгоритма в целом

1.4. Оценка выбранной метрики ПС.

2. На втором этапе выполняются следующие действия.

2.1. Анализ полученных результатов, поиск участков, оказывающих негативное влияние на эффективность стратегии кеширования

2.2. Коррекция алгоритма, получение нового разбиения $U \to U'$

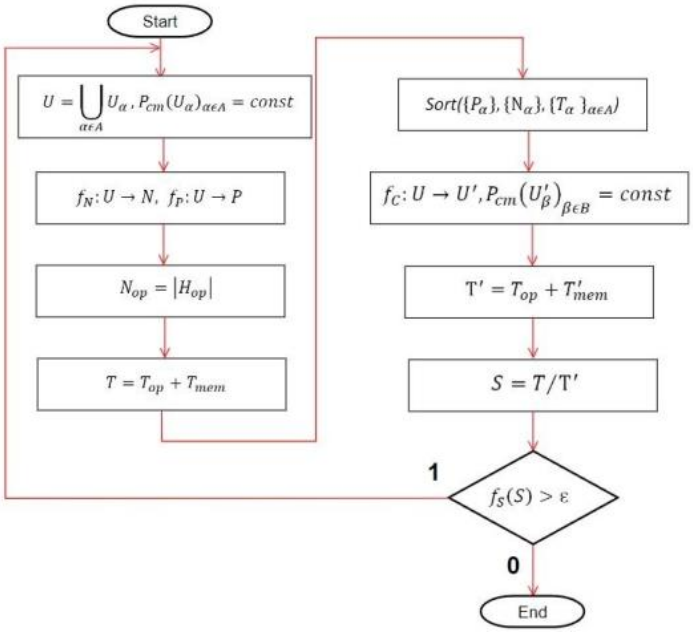


Рисунок 12 - Метод повышения эффективности использования ПП

2.3. Оценка нового значения выбранной метрики.

В данном методе используются следующие метрики ПС:

1. Ускорение программы в качестве метрики целесообразно использовать в том случае, если корректировка исходного алгоритма приводит к изменению числа арифметико-логических операций (на рисунке 12 в качестве метрики используется ускорение программы).

2. Ускорение работы с памятью в качестве метрики целесообразно использовать в том случае, если корректировка исходного алгоритма не приводит к изменению числа арифметико-логических операций, однако число операций обращения к памяти изменяется.

3. Усредненное значение кеш-промаха в виде метрики целесообразно использовать в случае, если корректировка исходного алгоритма не изменяет количество арифметико-логических операций и операций работы с памятью.

Метод повышения эффективности векторизации вычислений (метод ведущих векторов, МВВ, рисунок 13) позволяет существенно повысить эффективность ПС, в процессе выполнения которых осуществляется регулярный доступ к данным.

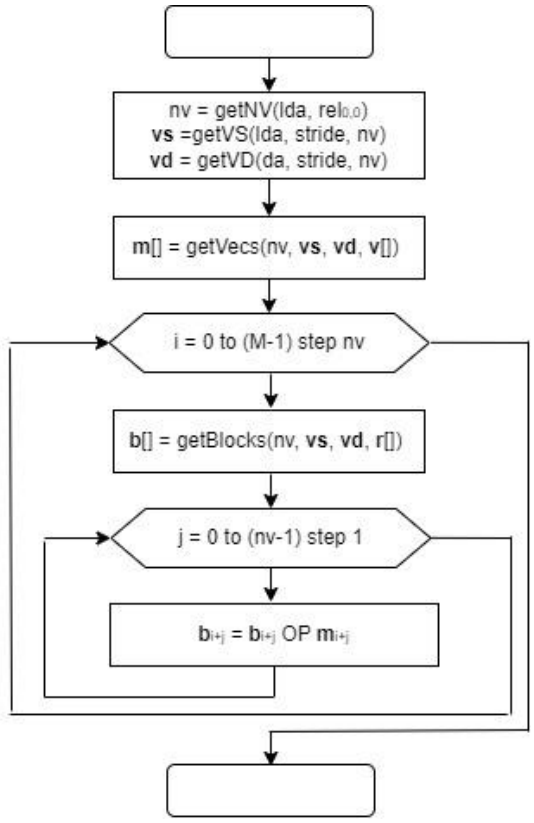
Применение МВВ позволяет избавиться от ограничений, присущих блокам автовекторизации современных оптимизирующих компиляторов:

– избежать формирования прологов и эпилогов за счет обработки только целых векторов;

– обеспечить одновременный выровненный доступ к нескольким массивам, обрабатываемым в цикле, что по умолчанию в общем случае не представляется возможным.

В МВВ используются следующие метрики ПС:

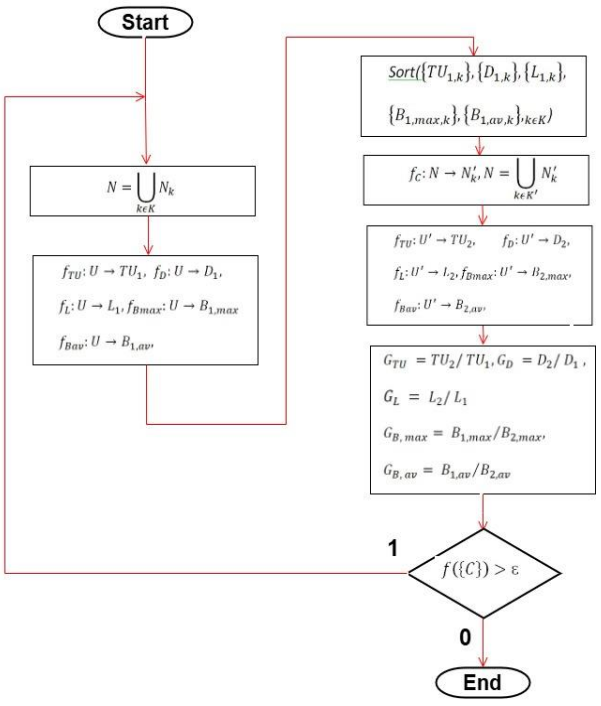
- увеличение потенциального ускорения;
- снижение доли невыровненных обращений.



1. Первый этап (формирование набора ВВ):
1.1. Определение размерности используемых векторов, их количества nv, смещения их начальных элементов vd, размерности обрабатываемых массивов данных в направлении обработки lda и других характеристик векторизуемого цикла;
1.2. Формирование выровненных ВВ в соответствии с определенными ранее параметрами m[];
1.3. Определение длин полученных векторов vs, смещений проекций векторов относительно обрабатываемых блоков данных r[].
2. Второй этап (изменение циклов доступа к данным):
2.1. Развертка цикла в соответствии с числом ведущих векторов;
2.2. Изменение параметров цикла (стартовый элемент, количество итераций и т.д.) в соответствии с параметрами ведущих векторов.

Рисунок 13 – Метод ведущих векторов

Метод повышения эффективности использования GPU (рисунок 14) позволяет повысить уровень загрузки и степень сбалансированности загрузки потоковых процессоров. Метод состоит из двух этапов.



1. Первый этап:
1.1. Разбиение множества операций N на вычислительные ядра (kernels) N_k задание параметров ядер $NB,\ NT_b$
1.2. Оценка значений параметров $TU_1,\ D_1,\ L_1,\ B_{1,\,max},\ B_{1,\,av}$ отдельно для каждого вычислительного ядра и на всем множестве операций
2. Второй этап:
2.1. Анализ полученных результатов, поиск участков, оказывающих негативное влияние на эффективность использования GPU
2.2. Коррекция алгоритма, получение нового разбиения $N \to N_k$
2.3. Оценка значений параметров $TU_2,\ D_2,\ L_2,\ B_{2,\,max},\ B_{2,\,av}$ отдельно для каждого вычислительного ядра и на всем множестве операций
2.4. Оценка метрик ПС $G_{TU},\ G_D,\ G_L,\ G_{B,\,max},\ G_{B,\,av}$

Рисунок 14 – Метод повышения эффективности использования GPU

В методе повышения эффективности использования GPU используются следующие метрики ПС:

- увеличение доли использованных потоков;
- увеличение средней плотности вычислений;

- увеличение средней степени локальности данных;

- увеличение степени сбалансированности потоков (максимальной и средней).

На базе разработанных моделей и методов была предложена методология проектирования эффективных ПС для решения инженерных задач в ГВС (рисунок 15).

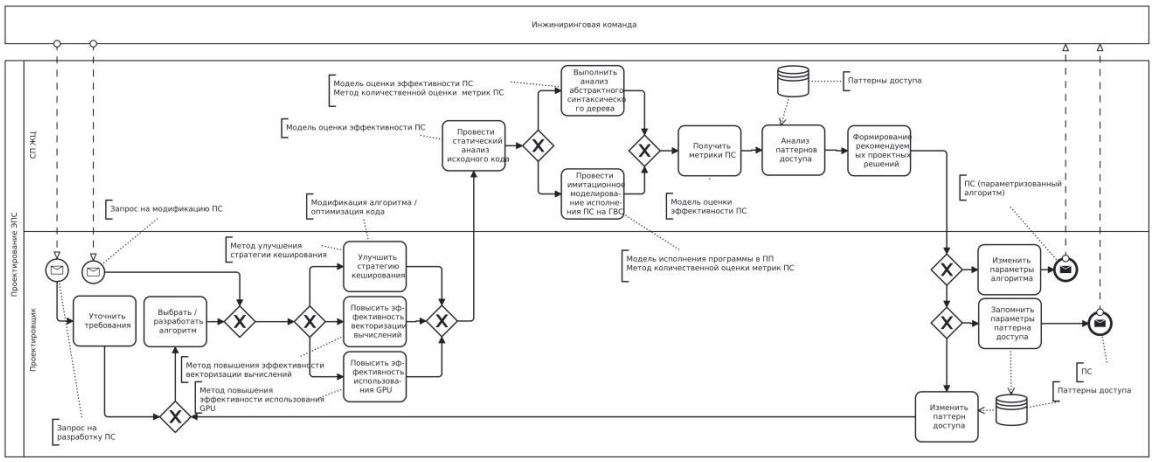


Рисунок 15– Схема методологии проектирования ЭПС

Данная методология базируется на взаимодействии проектировщика ПС с системой поддержки жизненного цикла ПС (СП ЖЦ) основанной на реализации предложенных моделей и методов. СП ЖЦ позволяет:

- выполнять аналитическую оценку метрик ПС;

- проводить имитационное моделирования исполнения ПС в ГВС;

- выбирать проектные решения по повышению эффективности ПС.

Оценка метрик ПС осуществляется путем поиска и формализации паттернов доступа в память:

- выявленные ранее, для которых ранее получены необходимые метрики; характеристики данных паттернов используются для анализа эффективности ПС;

- новые паттерны доступа, информация о которых отсутствует в системе; для них производится имитационное моделирование с целью определения метрик, полученные параметры запоминаются для повторного использования;

- в отдельных случаях для новых паттернов доступа метрики ПС могут быть определены аналитически.

Имитационное моделирование осуществляется в соответствии с диаграммой (рисунок 16). В процессе выполнения симуляции интерпретатор формирует последовательность трасс, содержащих адреса, по которым осуществляется обращения к памяти.

В отличие от традиционных трасс-ориентированных симуляторов, предложенная инженерная методика позволяет накапливать данные о выполняемом коде благодаря возможности передачи в трассах дополнительных параметров.

Так, трасса в общем случае задается в виде кортежа (21).

$$T = \langle t, \{P_i\} \rangle \quad (21)$$

где t - трасса (например, адрес, по которому произошло обращение); $\{P_i\}$- множество дополнительных параметров.

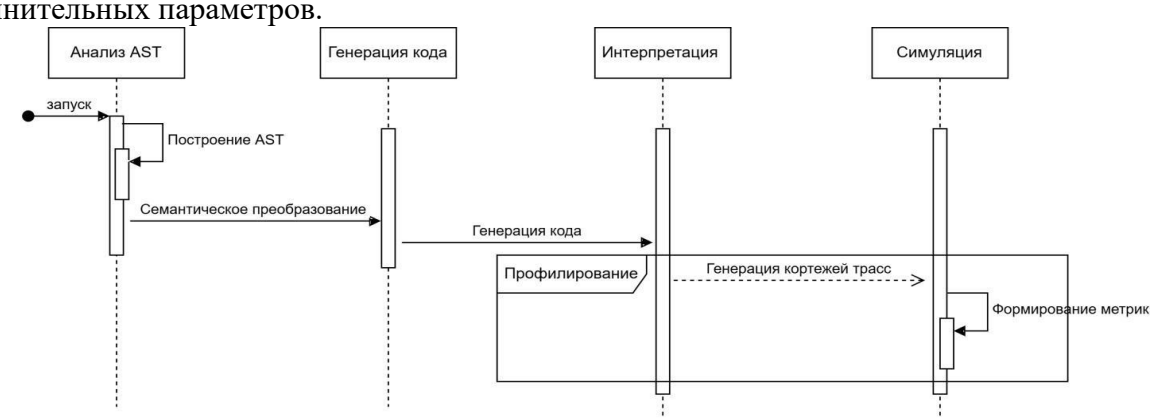


Рисунок 16 – Инженерная методика выполнения имитационного моделирования

Дополнительные параметры позволяют после выполнения имитационного моделирования не только получить значения метрик ПС, но и проанализировать то, каким образом они формируются: название массива, тип операции, имя функции (метода), имя класса и др.

В четвертой главе представлена верификация созданной методологии проектирования ЭПС на примере практически значимых вычислительно-сложных задач.

Методологический подход был реализован для задачи приведения матрицы общего вида к Хессенберговой форме (ХФ) методом Хаусхолдера (Basic Linear Algebra Subroutines, BLAS, уровень 3). Приведение исходной матрицы к ХФ является распространенным способом предварительной обработки матрицы в процессе решения задачи собственного разложения. Данное преобразование позволяет уменьшить время решения задачи собственного разложения, однако, само двустороннее преобразование Хаусхолдера в его традиционном классическом виде является неэффективным в ВС с иерархической структурой памяти.

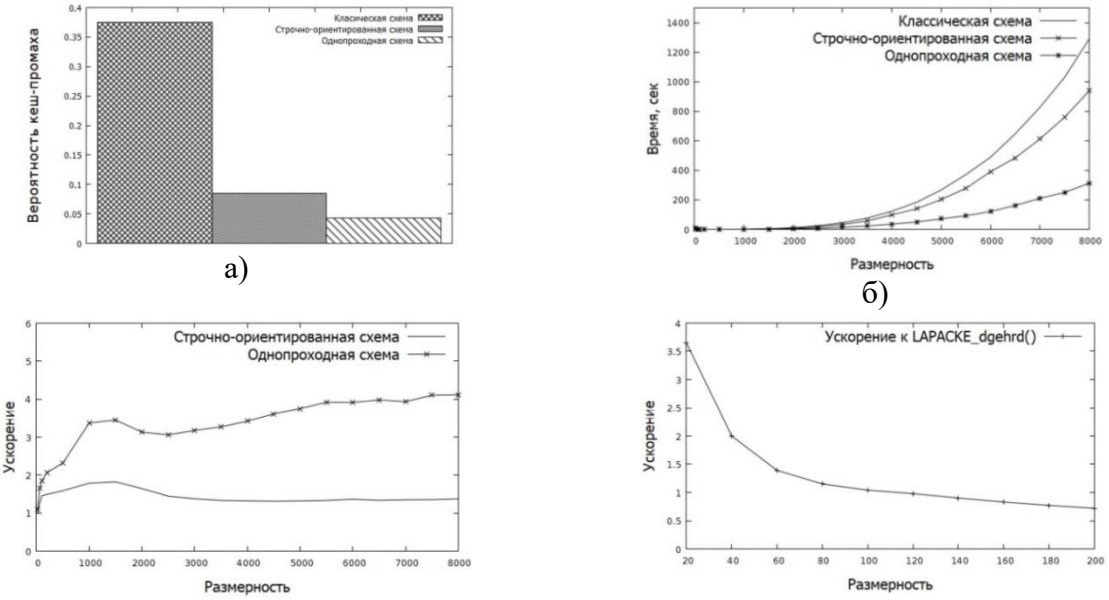
Процесс проектирования содержал три итерации проектных процедур, применены МУСК и МВВ (таблица 7).

Таблица 7 – Реализация методологического подхода для задачи приведения матрицы общего вида к ХФ методом Хаусхолдера

№ итерации	Реализуемый метод	Улучшаемые метрики	Полученная вычислительная схема
1	МУСК	R_P, S_P	Строчно-ориентированная
2	МУСК	R_P, S_P	Однопроходная
3	МВВ	S_P, R_U	Однопроходная векторизованная

На первых двух итерациях осуществлялось разбиение множества операций доступа к памяти на непересекающиеся подмножества, оценивались количество элементов и вероятности кеш-промахов в каждом подмножестве, выявлялись подмножества, оказывающие существенное негативное влияние на эффективность реализации исследуемых схем преобразования. Выполнялись перестановочные оптимизации, в результате чего были получены две новых схемы выполнения преобразования, которые позволили значительно улучшить

значения метрик ПС (рисунок 17). Эксперименты показали, что МУСК позволяет до 4 раз ускорить вычисления, до 3-4 раз снизить энергопотребление ПС.



а) б) в) г)

Рисунок 17 – Полученные метрики ПС: усредненное значение вероятности кеш-промаха (а), время работы (б), ускорение программы (в-г)

На третьей итерации был реализован МВВ, что позволило избежать потерь производительности при доступе к данным. На ряде размерностей было получено значительное ускорение по отношению к программной реализации, полученной с использованием Intel MKL, в 3,7 раза.

МВВ был реализован для задачи матрично-векторного умножения (BLAS 2) и задач обработки числовых последовательностей на примере задач вычисления одномерных и двумерных сверток (таблица 8). Процесс проектирования был реализован за одну итерацию проектных процедур.

Таблица 8 – Реализация методологического подхода к задачам матрично-векторного умножения и обработки числовых последовательностей

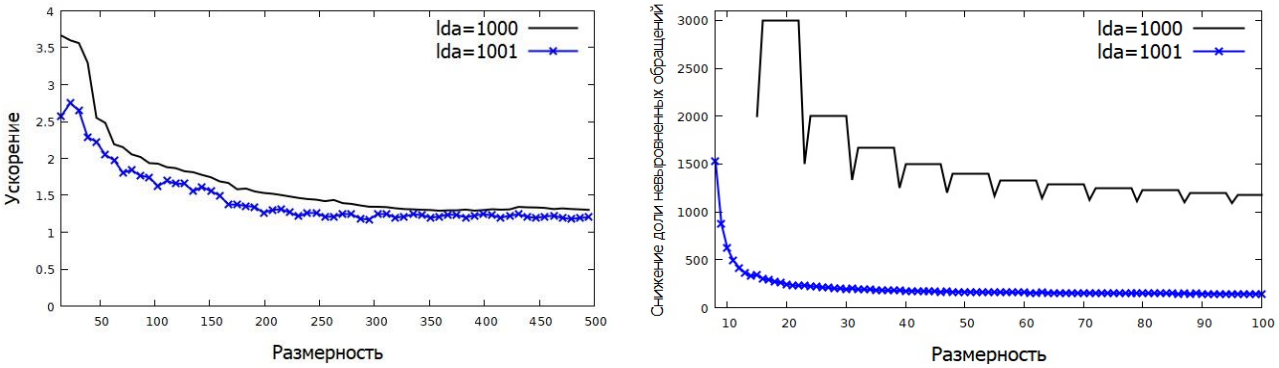
№ итерации	Реализуемый метод	Улучшаемые метрики	Полученное ускорение относительно результатов, полученных с помощью автоматической векторизации
1	МВВ	S_P, R_U	до 3,7 раз

Задача матрично-векторного умножения заключается в умножении подматрицы, привязанной к некоторому начальному элементу, на заданный вектор. В целом приведенное преобразование аналогично функции BLAS sgemv, т.к. вычисления выполнялись над числами с плавающей запятой одинарной точности (float).

На первом этапе были определены характеристики векторизуемого цикла и формирование необходимого числа ведущих векторов, определены их длины и смещения. Второй этап в основном связан с написанием программного кода: изменяются циклы доступа к данным в соответствии с полученными параметрами.

На рисунке 18а приведено полученное ускорение для различных ведущих размерностей (lda=1000, lda=1001) относительно реализаций, полученных с

использованием автоматической векторизации, на рисунке 18б приведена зависимость снижения доли невыровненных обращений от размерности решаемой задачи.



а) б)

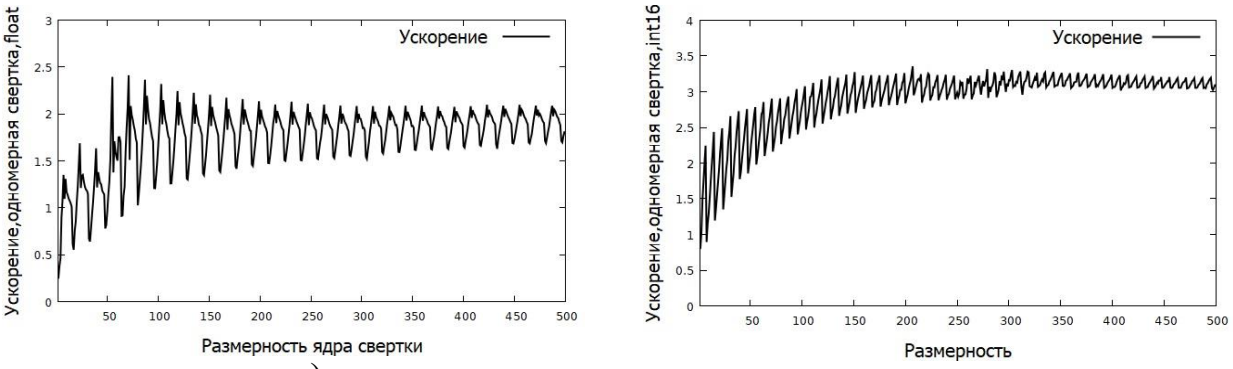
Рисунок 18 – МВВ: реализация метода (а), полученная метрика ПС (ускорение) (б)

Программа была написана на языке программирования C++, поэтому в качестве ведущей размерности массива выступала длина строки. Видно, что ускорение для lda=1001 меньше, чем для lda=1000. Это связано с тем, что в первом случае необходимо формировать большее число ведущих векторов.

Эксперименты показали, что применение МВВ значительно ускоряет работу проектируемой ПС. На небольших размерностях выигрыш составляет до 3,7 раз, с ростом размерности значение ускорения уменьшается и стабилизируется при достижении значения 1,2.

На рисунке 19 приведено ускорение, полученное при реализации методологического подхода к задаче вычисления одномерной свертки для различных типов данных относительно реализаций, полученных с использованием автоматической векторизации.

Зависимости, представленные на рисунке 19, имеют нелинейный характер. Это связано с тем, что в процессе автоматической векторизации компилятор разворачивает цикл, формирует прологи и эпилоги циклов, при этом время вычислений зависит от длин прологов и эпилогов циклов, которые зависят, в том числе, и от размерности ядра свертки.



а) б)

Рисунок 19 – МВВ: полученная метрика ПС (ускорение) для типа данных float (а) и __int16 (б)

В пятой главе представлены концепция и архитектура системы поддержки ЖЦ проектируемых ПС (рисунок 20), реализующей предложенные модели и методы. Система состоит из трех основных подсистем.

1. Подсистема формирования характеристик кода. Выполняет статический анализ исходного кода:

- осуществляет поиск паттернов доступа к данным (регулярный / нерегулярный доступ, непрерывно расположенные данные / расположение данных с интервалами), определение параметров регулярных паттернов доступа.
- выполняет имитационное моделирование исполнения программы, интерпретирующее профилирование, определение параметров нерегулярных паттернов доступа, паттернов с длиной, превышающей размер анализируемого уровня памяти ГВС.

Реализация подсистемы основана на модели улучшения показателей эффективности ПС, математической модели оценки эффективности проектируемой ПС, аналитическом методе определения максимального размера блока при регулярном доступе к данным, имитационной модели исполнения программы в ГВС с иерархической структурой памяти.

2. Подсистема оценки эффективности кода. Формирует метрики проектируемой ПС.

Реализация подсистемы основана на математической модели оценки эффективности проектируемой ПС, аналитическом методе определения максимального размера блока при регулярном доступе к данным, комплексе методов повышения эффективности ПС (МУСК, МВВ, методе повышения эффективности использования GPU).

3. Подсистема поддержки проектирования ПС.

Определяются параметры алгоритмов.

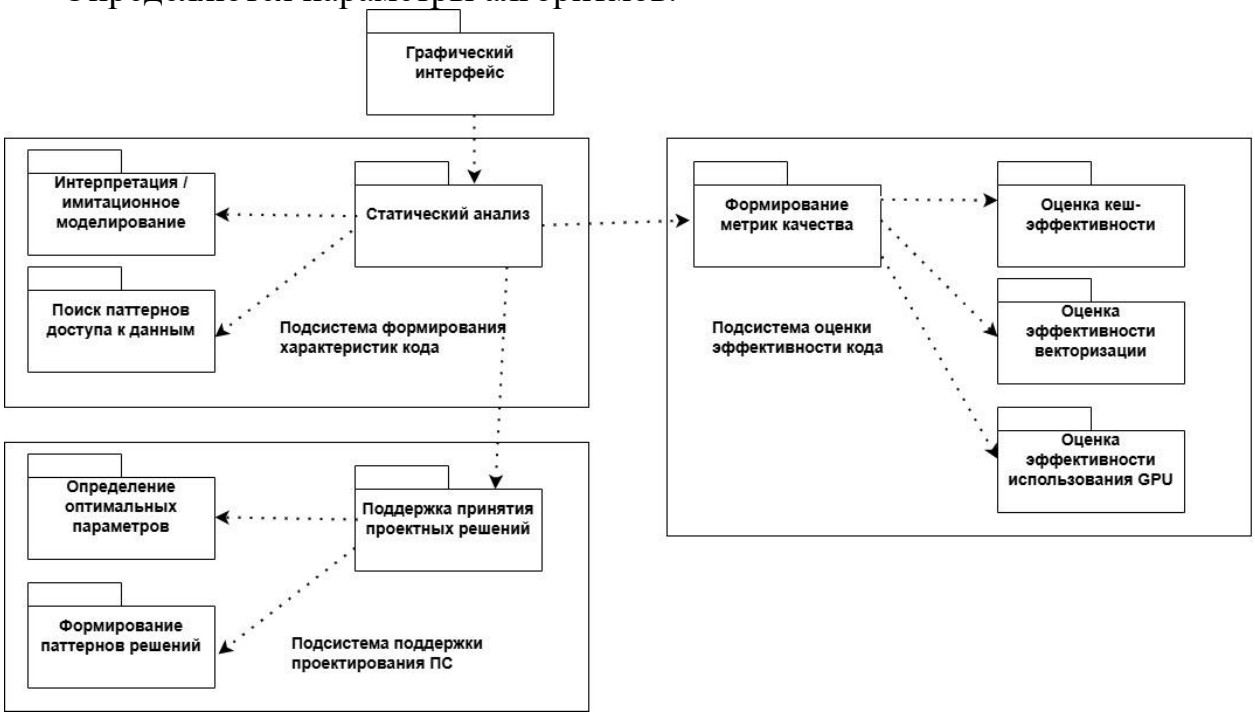


Рисунок 20 - Архитектура системы поддержки ЖЦ проектируемых ПС

Реализация подсистемы основана на комплексе методов повышения эффективности ПС, методологии проектирования ЭПС для решения инженерных задач в ГВС.

На рисунке 21 представлена экранная форма подсистемы формирования характеристик кода системы поддержки ЖЦ.

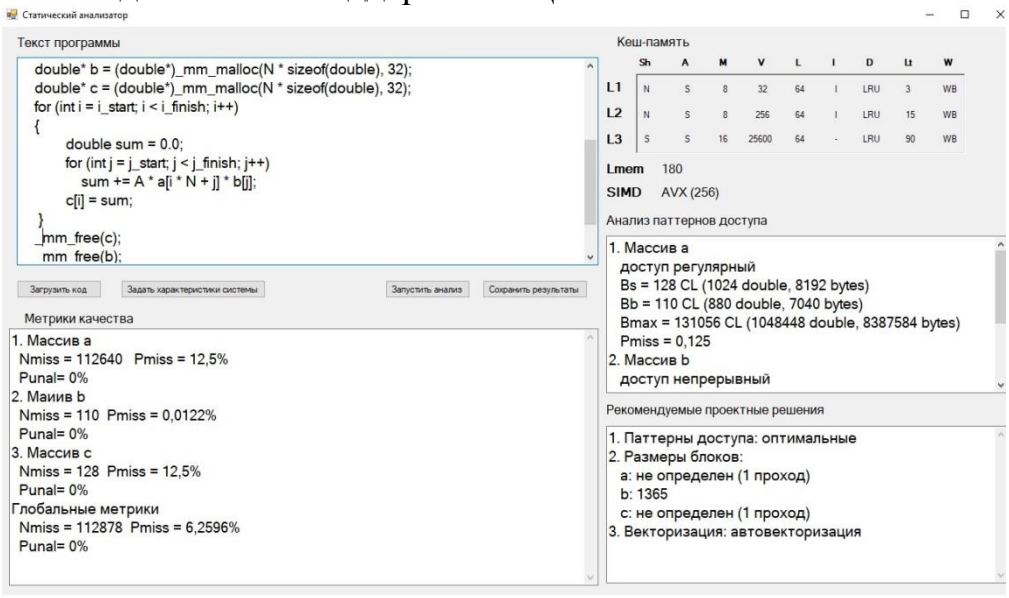


Рисунок 21 - Экранная форма подсистемы формирования характеристик кода

Реализация предложенного методологического подхода позволила значительно повысить эффективность проектируемых ПС на различных этапах ЖЦ ПС (таблица 9).

Таблица 9 – Результаты реализации научно-методологического подхода

Параметр	**Этап ЖЗ**	**Результат**
Время разработки (модификации) ПС	Разработка, модификация ПС	Ресурсы, затрачиваемые на разработку и модификацию ПС, снижены за счет значительного (в 2-3 раза) снижения числа итераций проектных процедур.
Время работы программы	Эксплуатация ПС	Время решения вычислительно- сложных задач снижено в результате применения МУСК (BLAS 3) до 4 раз, в результате применения МВВ (BLAS 2) снижено в 3,7 раза.
Ускорение программы	Эксплуатация ПС	Ускорение, полученное при решении задач BLAS 3 в результате применения МУСК, увеличивается с ростом размерности задачи, ускорение, полученное в результате применения МВВ, достигая значения 3,7, уменьшается с ростом размерности задачи и стабилизируется на значении 1,2.
Энергоэффективность	Эксплуатация ПС	Применение предложенных моделей и методов позволяет до 3-4 раз сократить энергопотребление ПС.

Программные решения, реализованные в системе поддержки ЖЦ, были использованы в процессе создания и оптимизации программного обеспечения, разрабатываемого компанией «ТЕСИС» для моделирования гидроаэродинамики и тепловых процессов, а также решения междисциплинарных задач, в частности – при разработке прототипов расчетных библиотек.

Разработанные алгоритмы использовались для выполнения базовых блочных операций линейной алгебры (умножение матриц, обращение и факторизация матриц) для разреженных мелко-блочных матриц и блоков векторов (плотных матриц определенной размерности) с применением векторных расширений системы команд микропроцессоров Intel и использованием возможностей графических сопроцессоров. На основе базовых операций компанией «ТЕСИС» были разработаны алгоритмы умножения разреженных мелко-блочных матриц на блоки векторов, LU-разложения, обращения матриц и решения систем линейных алгебраических уравнений. Реализация разработанных алгоритмов и методов позволило получить значительное ускорение выполнения базовых операций до 10 раз относительно предыдущих программных реализаций за счет векторизации вычислений. Разработанные алгоритмы и методы, использующие графические сопроцессоры, позволяют значительно, до 50-75%, уменьшить время вычислений над плотными и разреженными матрицами за счет распределения вычислений между центральным процессором и GPU, оптимизации вычислений под конкретные платформы, интеграции ПС с существующими высокопроизводительными программными библиотеками. Кроме того, полученные наработки в области векторизации вычислений и распределения вычислений между центральным процессором и GPU компания «ТЕСИС» предполагает использовать при построении вычислительных библиотек для программного комплекса «FlowVision» в 2025 – 2027 г.

Программные решения, реализованные системе поддержки ЖЦ ПС, были использованы в ЗАО «Топ Системы» в инструментальных компонентах, разработанных для системы управления полным жизненным циклом изделий для промышленных предприятий «САРУС». Разработанные методы и алгоритмы вычислительно-сложных процедур использованы в библиотеках классов, предназначенных для решения геометрических задач в прототипе CAD-системы ядра 3-мерного моделирования: построения эквидистантных линий к проволочному телу; реализации операций вычисления масс-инерционных характеристик проволочных, твердых и листовых тел с использованием интегрирования по сетке; оптимизации ряда базовых расчетных классов библиотеки геометрических операций, связанных с реализацией булевых операций с телами; выполнения кинематических операций, построения пересечений и поиска самопересечений и оптимизации частных случаев обрезки кривых и поверхностей. Использование указанных результатов позволило повысить эффективность ПС: уменьшить время, необходимое для реализации базовых расчетных процедур до 18%, повысить точность вычислений до 7%. Годовой экономический эффект, полученный от внедрения результатов диссертационной работы, оценивается в 1,8 млн. рублей.

Программные решения, реализованные в системе поддержки ЖЦ, были использованы при реализации ПС для сбора и обработки данных с производственных участков электроплавильного цеха (ЭПЦ), а также для многокритериального анализа факторов, влияющих на объем и качество выпускаемой продукции в ООО «Волжский абразивный завод». Использование

указанных решений позволило повысить эффективность сбора и обработки данных ЭПЦ на 17-20%, снизить количество ошибок, обусловленных человеческим фактором на 7%. Годовой экономический эффект, полученный от внедрения результатов диссертационной работы, оценивается в 1 млн. рублей.

В **Заключении** описаны основные результаты работы, выводы и перспективы исследования.

В **Приложениях** приведены документы, подтверждающие апробацию и внедрение результатов.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Главным результатом работы являются научно-методологические основы поддержки жизненного цикла проектируемых программных систем для решения инженерных задач в гетерогенных вычислительных средах, повышающие эффективность программных систем на различных этапах жизненного цикла.

Основные результаты, в которых отражена научная новизна и практическая ценность диссертационной работы, заключаются в следующем:

1. Исследованы существующие модели, методы и CASE-средства, использующиеся для проектирования ПС в области высокопроизводительных вычислений. Рассмотрены полные тексты более 40 публикаций, часть из этих публикаций представляют собой систематические исследования с общим числом исследованных работ более 600. Выявлены основные тенденции в области разработки моделей проектирования ПС, методов, основанных на этих моделях, выявлены основные недостатки данных моделей и методов.

2. Предложена модель улучшения показателей эффективности ПС, которая позволяет сократить время проектирования. Применение данной модели снижает затраты на разработку и модификацию ПС за счет значительного (в 2-3 раза) уменьшения числа итераций проектных процедур.

3. Предложена математическая модель оценки эффективности проектируемых ПС, позволяющая проектировщику ПС комплексно оценить ПС на различных этапах ЖЦ в ГВС.

4. Предложен аналитический метод определения максимального размера блока при регулярном доступе к данным, позволяющий производить исследование показателей эффективности использования подсистемы памяти проектируемых ПС. Сформулированы и доказаны 4 теоремы, формально описывающие отображение множества кешируемых данных на множества кеш-памяти. Сформулирован ряд следствий из теорем, позволяющих аналитически оценить степень заполняемости множеств кеш-памяти с различными характеристиками при организации регулярного доступа к данным на основе различных шаблонов доступа. Реализация данного метода в процессе анализа ПС позволяет выбирать параметры алгоритмов, минимизирующие количество кеш-промахов при доступе к данным, тем самым значительно ускоряя работу ПС.

5. Предложена имитационная модель исполнения программы в ГВС с иерархической структурой памяти, позволяющая определять характеристики ПС на конкретной целевой вычислительной архитектуре, а также характеристики подсистемы памяти для проектируемой ПС. Реализация модели дает

проектировщику ПС возможность проведения имитационного моделирования процесса выполнения программы в подсистеме памяти, что позволяет оценивать ПС и принимать проектные решения по повышению эффективности ПС.

6. Предложен комплекс методов, позволяющий значительно повысить эффективность проектируемых ПС за счет значительного (до 4 раз) снижения времени экспериментальных вычислений и уменьшения энергопотребления ПС в 3-4 раза на этапе эксплуатации, снижения затрат на проектирование ПС:

– метод повышения эффективности использования подсистемы памяти, позволяющий улучшать стратегию кеширования данных в ГВС с иерархической структурой памяти;

– метод повышения эффективности векторизации вычислений, позволяющий избежать потерь производительности при доступе к данным за счет отказа от невыровненных обращений в память;

– метод повышения эффективности использования GPU, позволяющий повысить уровень загрузки и степени сбалансированности загрузки потоковых процессоров, что позволило увеличить среднюю плотность вычислений, снизить расхождение ветвей, увеличить долю использованных потоков.

7. Предложена методология проектирования эффективных ПС для решения инженерных задач в ГВС, реализующая предложенные модели и методы, позволяющая принимать проектные решения по эффективному использованию вычислительных ресурсов на начальных этапах жизненного цикла ПС. Реализация данной методологии позволяет повысить эффективность работы проектировщика, эффективность проектируемых ПС.

8. Выполнена верификация созданной методологии проектирования ЭПС на примере практически значимых вычислительно-сложных задач.

9. Предложены концепция и архитектура системы поддержки ЖЦ проектируемых ПС, позволяющей автоматизировать анализ эффективности возможных проектных решений о стратегиях разработки и модификации ПС, которая состоит из трех основных подсистем: подсистема формирования характеристик кода, подсистема оценки эффективности кода, подсистема поддержки проектирования ПС.

В качестве перспектив исследования можно выделить развитие методологии проектирования ЭПС за счет анализа параллельных и распределенных вычислений, расширения перечня анализируемых вычислительных узлов в составе ГВС, а также развитие системы поддержки ЖЦ проектируемых ПС.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в изданиях, включенных в перечень ВАК при Министерстве образования и науки РФ по специальностям 2.3.7 и 05.13.12

1. Егунов, В.А. Новый подход к проектированию эффективных программных систем для высокопроизводительных вычислений на основе анализа жизненного цикла // Информационные технологии. - 2025. - Т. 31, № 7. - С. 379-392. - DOI: 10.17587/it.31.379-392.

2. Егунов, В.А. Определение показателей эффективности программных систем с регулярным доступом к данным для вычислительных сред с

иерархической структурой памяти / В.А. Егунов // Вестник компьютерных и информационных технологий. - 2025. - Т. 22, № 6. - С. 3-11. - DOI: 10.14489/vkit.2025.06.pp.003-011.

3. Егунов, В.А. Новый метод определения характеристик блочных алгоритмов / Вестник Дагестанского государственного технического университета. Технические науки. - 2025. - Т. 52, № 2. - С. 90-97. DOI: 10.21822/2073-6185-2025-52-2-90-97 .

4. Егунов, В.А. Использование методов интерпретации и компиляции для повышения эффективности программного обеспечения / В.А. Егунов, В.А. Шабаловский // Цифровая экономика. - 2024. - № 3 (29). - С. 65-71.

5. Исследование эффективности методов компрессии данных в реляционных и NoSQL СУБД / В.А. Егунов, В.С. Сурин, П.С. Ступницкий, Р.Д. Ахмедова // Вестник Дагестанского государственного технического университета. Технические науки. - 2024. - Т. 51, № 1. - С. 87-94. - DOI: 10.21822/2073-6185-2024-51-1-87-94. - URL: https://vestnik.dgtu.ru/jour/article/view/1458/863.

6. Егунов, В.А. Новый метод повышения эффективности векторизации операций BLAS / В.А. Егунов, А.Г. Кравец // Информационные технологии. - 2024. - Т. 30, № 6. - С. 318-328. - DOI: 10.17587/it.30.318-328.

7. Егунов, В.А. Оценка эффективности параллельных программ с использованием Intel Parallel Studio / В.А. Егунов, В.А. Шабаловский // Информатизация и связь. - 2024. - № 1. - С. 118-123. - DOI: 10.34219/2078-8320-2024-15-118-123.

8. Егунов, В.А. Технологии кеширования данных в современных микропроцессорах / В.А. Егунов, В.А. Шабаловский // Вестник Дагестанского государственного технического университета. Технические науки. - 2024. - Т. 51, № 3. - С. 60-71. - DOI: https://doi.org/10.21822/2073-6185-2024-51-3-60-71.

9. Егунов, В.А. Анализ средств контроля повышения энергоэффективности программного обеспечения / В.А. Егунов, Д.Л. Абдрахманов // Цифровая экономика. - 2023. - № 2 (23). - С. 65-70. - DOI: 10.34706/DE-2023-02-08.

10. Безрученко, А.Ю. Исследование эффективности методов оптимизации программ для параллельных вычислительных систем с GPU / А.Ю. Безрученко, В.А. Егунов // Вестник Дагестанского государственного технического университета. Технические науки. - 2023. - Т. 50, № 4. - С. 59-74. – DOI: 10.21822/2073-6185-2023-50-4-59-74. – URL: https://vestnik.dgtu.ru/jour/article/view/1392/842.

11. Егунов, В.А. Реализация геометрического преобразования отражения для прототипа высокопроизводительного геометрического ядра / В.А. Егунов, О.Ю. Филимонов // Цифровая экономика. - 2023. - № 1 (22). - С. 20-26. - DOI: 10.34706/DE-2023-01-03.

12. Чекушкин, А.А. Эффективная реализация сингулярного разложения на вычислительных системах с общей памятью / А.А. Чекушкин, В.А. Егунов, А.Г. Кравец // Информатизация и связь. - 2023. - № 1. - С. 112-119. - DOI: 10.34219/2078-8320-2023-14-1-112-119.

13. Егунов, В.А. О влиянии кэш-памяти на эффективность программной реализации базовых операций линейной алгебры / В.А. Егунов // Прикаспийский журнал: управление и высокие технологии. - 2018. - № 3. - С. 88-96.

14. Егунов, В.А. Разработка инерциальной системы навигации шагающего робота с возможностью визуализации положения робота в пространстве / В.А. Егунов, М.К. Петросян // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. - Волгоград, 2018. - № 5 (215). - С. 106-109

15. Абдрахманов, Д.Л. Система мониторинга вычислительного кластера / Д.Л. Абдрахманов, В.А. Егунов // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. - Волгоград, 2018. - № 13 (223) Декабрь. - С. 108-112.

16. Оценка эффективности программной реализации QR-разложения на многоядерных архитектурах / В.А. Егунов, С.И. Кирносенко, П.Д. Кравченя, О.О. Шумейко // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. - Волгоград, 2017. - № 1 (196). - С. 56-59.

17. Применение векторных инструкций в алгоритмах блочных операций линейной алгебры / А.Е. Андреев, В.А. Егунов, А.А. Насонов, А.А. Новокщенов // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 21 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2014. - № 12 (139). - С. 5-11.

18. Егунов, В.А. Об использовании линейных преобразований в управлении мобильным роботом / В.А. Егунов, С.В. Артюх // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 17 : межвуз. сб. науч. тр. / ВолгГТУ. - Волгоград, 2013. - № 14 (117). - С. 82-84.

19. Егунов, В.А. Трехуровневая архитектура мобильных робототехнических комплексов / В.А. Егунов, М.И. Потапов // Известия ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах». Вып. 13 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2012. - № 4 (91). - С. 159-161

20. Егунов, В.А. Алгоритм направленного дискретного линейного преобразования отражения / В.А. Егунов // Изв. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 12 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2011. - № 11. - С. 47-49.

Статьи в изданиях, индексируемых в Scopus и WoS

21. Egunov V. A., Kravets A. G.. A Method for Improving the Caching Strategy for Computing Systems with Shared Memory // Programming and Computer Software. - 2024. - Vol. 50, Issue 2 supplement (December). – P. S113-S119. – DOI: https://doi.org/10.1134/S036176882470049X.

22. Egunov V. A., Kravets A. G.. The New Method for Automatic Vectorization Efficiency Increasing // Cyber-Physical Systems. Data Science,

Modelling and Software Optimization / eds.: A. G. Kravets, A. A. Bolshakov. – Cham (Switzerland) : Springer Nature Switzerland AG, 2024. – P. 195-208. – DOI: https://doi.org/10.1007/978-3-031-67685-7_14. – (Book ser.: Studies in Systems, Decision and Control (SSDC) ; vol. 554).

23. Egunov, V., Kravets, A.G., The Method for Increasing the Software Efficiency for Computing Systems with a Hierarchical Memory Structure / Cyber-Physical Systems Engineering and Control / eds.: A. G. Kravets, A. A. Bolshakov, M. V. Shcherbakov. – Cham (Switzerland) : Springer Nature Switzerland AG, 2023. – P. 221-231. – DOI: https://doi.org/10.1007/978-3-031-33159-6_17. – (Book ser.: Studies in Systems, Decision and Control (SSDC) ; vol. 477)

24. Kravets, A.G., Egunov, V. The Software Cache Optimization-Based Method for Decreasing Energy Consumption of Computational Clusters // Energies. - 2022. - Vol. 15, issue 20 (October-2) [Special issue «Smart Energy and Sustainable Environment»]. – Article 7509. – 16 p. – DOI: https://doi.org/10.3390/en15207509

25. The problem of applicability of computer-aided design systems in the production of elements of hydraulic units / E. N. Nesterenko, P. S. Nesterenko, J. L. Tchigirinsky, V. A. Egunov // Journal of Physics: Conference Series, Volgograd, Virtual, 20–21 октября 2020 года. – Volgograd, Virtual, 2021. – P. 012002. – DOI 10.1088/1742-6596/1801/1/012002.

26. Filimonov, O. Y. Constructing Equidistant Curve for Planar Composite Curve in CAD Systems / O. Y. Filimonov, V. A. Egunov, E. N. Nesterenko // Communications in Computer and Information Science. – 2021. – Vol. 1448. – P. 296-309. – DOI 10.1007/978-3-030-87034-8_22. – EDN TPJDRC.

27. Andreev, A.E. Egunov, V.A. Solving of Eigenvalue and Singular Value Problems via Modified Householder Transformations on Shared Memory Parallel Computing Systems. In Supercomputing: RuSCDays, Proceedings of the 5th Russian Supercomputing Days, Moscow, Russia, 23–24 September 2019; Springer: Cham, Switzerland, 2019; Volume 1129, pp. 131–151.

28. Andreev, A.E. Chalyshev M.S., Egunov, V.A., Doukhnitch E.I., Kuznetsova K. Effective Quaternion and Octonion Cryptosystems and Their FPGA Implementation // Creativity in Intelligent Technologies and Data Science (CIT&DS 2019) : Third Conference (Volgograd, Russia, September 16–19, 2019) : Proceedings. Part I / Editors: A. Kravets, P. Groumpos, M. Shcherbakov, M. Kultsova ; Volgograd State Technical University [et al.]. – Cham (Switzerland) : Springer Nature Switzerland AG, 2019. – P. 406-419. – (Ser. Communications in Computer and Information Science (CCIS) ; Volume 1083)

29. Andreev, A.E., Egunov, V.A., Movchan E., Cherednikov N., Kharkov E., Kohtashvili N. The Introduction of Multi-level Parallelism Solvers in Multibody Dynamics // Creativity in Intelligent Technologies and Data Science (CIT&DS 2019) : Third Conference (Volgograd, Russia, September 16–19, 2019) : Proceedings. Part II / Editors: A. Kravets, P. Groumpos, M. Shcherbakov, M. Kultsova ; Volgograd State Technical University [et al.]. – Cham (Switzerland) : Springer Nature Switzerland AG, 2019. – P. 166-180. – (Ser. Communications in Computer and Information Science (CCIS) ; Volume 1084).

30. Glinsky, B.; Kulikov, I.; Chernykh, I.; Weins, D.; Snytnikov, A.; Nenashev, V.; Andreev, A.; Egunov, V.; Kharkov, E. The Co-design of Astrophysical Code for Massively Parallel Supercomputers. In Proceedings of the Algorithms and Architectures for Parallel Processing, ICA3PP 2016 Collocated Workshops: SCDT, TAPEMS, BigTrust, UCER, DLMCS, Granada, Spain, 14–16 December 2016; Lecture Notes in Computer Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; Volume 10049, pp. 342–353.

31. Egunov, V.A., Andreev, A.E. Implementation of QR and LQ decompositions on shared memory parallel computing systems. 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) (Chelyabinsk, Russia, 19-20 May 2016). – [Publisher: IEEE], 2016. – 5 p. – DOI: 10.1109/ICIEAM.2016.7911607.

32. Egunov, V.A., Kirnosenko S.I, Andreev, A.E. Robotic Complex Central Processing Node Performance Requirements Assessment // World Applied Sciences Journal (WASJ). - 2013. - Vol. 24, Spec. Issue 24 : Information Technologies in Modern Industry, Education & Society. - P. 37-42

33. Kirnosenko S.I, Egunov V.A., Andreev A.E., Zharikov D.N. Software Defect Prediction Based on Source Code Stabilization Model // World Applied Sciences Journal (WASJ). - 2013. - Vol. 24, Spec. Issue 24 : Information Technologies in Modern Industry, Education & Society. - P. 80-85

Монографии

34. Андреев А.Е., Егунов В.А., Жариков Д.Н., Малолетков В.А. Реализация вычислительно-интенсивных алгоритмов на гибридных системах с реконфигурируемыми сопроцессорами: монография / ВолгГТУ. - Волгоград, 2013. - 179 с.

35. Егунов, В.А., Лукьянов В.С. Аппаратные методы решения задач линейной алгебры: монография / ВолгГТУ. - Волгоград, 2007. - 152 с.

Свидетельства об официальной регистрации программы для ЭВМ

36. Свид. о гос. регистрации программы для ЭВМ № 2025662909 от 23 мая 2025 г. Российская Федерация. Модуль расчета метрик качества программы на языке С / В.А. Егунов; ФГБОУ ВО ВолгГТУ. - 2025.

37. Свид. о гос. регистрации программы для ЭВМ № 2025664671 от 05 июня 2025 г. Российская Федерация. Модуль двухуровневой адаптивной системы с цифровым двойником / В.А. Егунов, А.Г.Кравец, А.А.Кузьменко; ФГБОУ ВО ВолгГТУ. - 2025.

38. Свид. о гос. регистрации программы для ЭВМ № 2024667441 от 24 июля 2024 г. Российская Федерация. Программа для вычисления сингулярного разложения симметричной матрицы с использованием CUDA / А.Ю. Безрученко, В.А. Егунов, А.Г. Кравец; ФГБОУ ВО ВолгГТУ. - 2024.

39. Свид. о гос. регистрации программы для ЭВМ № 2023664318 от 3 июля 2023 г. Российская Федерация. Модуль поддержки векторизации матричных преобразований / В.А. Егунов, А.Г. Кравец; ФГБОУ ВО ВолгГТУ. - 2023.

40. Свид. о гос. регистрации программы для ЭВМ № 2023689295 от 27 декабря 2023 г. Российская Федерация. Программа обработки массивов данных

случайных величин перевозочного процесса грузов и пассажиров с расчетом параметров закона Эрланга / А.В. Куликов, И.Э. Симонова, В.А. Егунов, П.А. Павлов, А.А. Куликов; ФГБОУ ВО ВолгГТУ. - 2023.

41. Свид. о гос. регистрации программы для ЭВМ № 2014618996 от 5 сент. 2014 г. Российская Федерация. Автоматизированная система расчёта подачи резца при продольном точении нежёстких валов на станках с ЧПУ / А.А. Жданов, А.Л. Плотников, Ю.Л. Чигиринский, В.А. Егунов; ВолгГТУ. - 2014.

42. Свид. о гос. регистрации программы для ЭВМ № 2012614836 от 30 мая 2012 г. Российская Федерация. Программа управления сменными модулями масштабируемой архитектуры мобильного робота / В.А. Егунов, М.И. Потапов; ВолгГТУ. - 2012.

Прочие публикации

43. Егунов, В.А. Программные методы оценки энергопотребления вычислительных систем / В.А. Егунов, А.Г. Кравец, А.А. Чекушкин // Математические методы в технологиях и технике. - 2024. - № 12-1. - С. 50-54.

44. Егунов, В.А. Модели и методы проектирования программных систем для гетерогенных вычислительных сред / В.А. Егунов // Математические методы в технологиях и технике. - 2024. - № 11. - С. 133-140.

45. Егунов, В.А. Метод улучшения стратегии кеширования для вычислительных систем с общей памятью / В.А. Егунов, А.Г. Кравец // Программная инженерия. - 2023. - Т. 14, № 7. - С. 329-338. - DOI: 10.17587/prin.14.329-338.

46. Егунов В.А., Кравец А.Г. Повышение эффективности векторизации вычислений // Математические методы в технологиях и технике. - 2023. - № 3. - С. 65-68. - DOI: 10.52348/2712-8873_MMTT_2023_3_65.

47. Егунов В.А. , Кравец А.Г. Повышение эффективности программ для вычислительных систем с иерархической структурой памяти // Математические методы в технологиях и технике. - 2022. - № 4. - С. 100-103. – DOI: 10.52348/2712-8873_MMTT_2022_4_100.

48. Егунов, В.А. Векторизация алгоритмов выполнения собственного и сингулярного разложений матриц с использованием преобразования Хаусхолдера / В.А. Егунов, А.Е. Андреев // Прикаспийский журнал: управление и высокие технологии. - 2020. - № 2 (50). - С. 71-85.

49. Завьялов, Д.В. О векторизации алгоритма Монте-Карло решения классического уравнения Больцмана / Д.В. Завьялов, В.А. Егунов, В.И. Конченков // Математическая физика и компьютерное моделирование. - 2020. - Т. 23, № 1. - С. 13-21

50. Егунов, В.А. Кэш-оптимизация процесса вычисления собственных значений на параллельных вычислительных системах / В.А. Егунов // Прикаспийский журнал: управление и высокие технологии. - 2019. - № 1 (45). - С. 154-163.

51. Построение программной траектории движения беспилотного наземного транспортного средства / В.А. Егунов, А.Е. Марков, Ан.В. Скориков,

П.С. Тарасов // Прикаспийский журнал: управление и высокие технологии. - 2019. - № 3 (47). - С. 70-82

52. Optimization of Small Dimension Matrices Processing in a Multi-body System Modeling / A. Andreev, V. Egunov, A. Gorobtsov, E. Kharkov // Параллельные вычислительные технологии (ПаВТ'2021) : Короткие статьи и описания плакатов. XV международная конференция, Волгоград, 30 марта – 01 2021 года. – Челябинск: Издательский центр ЮУрГУ, 2021. – P. 29-41.

53. Андреев А.Е., Егунов В.А., Завьялов Д.В., Жариков Д.Н. О применении высокопроизводительных вычислений в фундаментальных исследованиях, прикладных и образовательных проектах ВолгГТУ // Параллельные вычислительные технологии – XV : международная конференция ПаВТ'2021 (г. Волгоград, 30 марта – 1 апреля 2021 г.) : короткие статьи и описания плакатов / РАН, Суперкомпьютерный консорциум университетов России, РФФИ. - Челябинск, 2021. - С. 131-142. – URL: http://omega.sp.susu.ru/pavt2021/proceedings.pdf.

54. Андреев А.Е., Егунов В.А., Завьялов Д.В., Жариков Д.Н. Развитие направления параллельных и высокопроизводительных вычислений в ВолгГТУ // Параллельные вычислительные технологии – XV : международная конференция ПаВТ'2021 (г. Волгоград, 30 марта – 1 апреля 2021 г.) : короткие статьи и описания плакатов / РАН, Суперкомпьютерный консорциум университетов России, РФФИ. - Челябинск, 2021. - С. 151-161. – URL: http://omega.sp.susu.ru/pavt2021/proceedings.pdf.

55. Andreev, A. Optimization of Two-Sided Householder Reflection Transformation for Shared Memory Parallel Computer Systems / A. Andreev, V. Egunov // Параллельные вычислительные технологии (ПаВТ'2020) : Короткие статьи и описания плакатов, Пермь, 31 марта – 02 апреля 2020 года. – Пермь: Издательский центр ЮУрГУ, 2020. – P. 8-17. – EDN SAXDLS.

56. Андреев А.Е., Егунов В.А. Векторизация алгоритмов выполнения собственного и сингулярного разложений с использованием преобразования // Национальный Суперкомпьютерный Форум (НСКФ-2019) (г. Переславль-Залесский, 26-29 ноября 2019 г.) : сб. тез. докл. НСКФ’2019. Секция «Прикладное программное обеспечение» / АНО «Национальный суперкомпьютерный форум», Ин-т программных систем им. А. К. Айламазяна РАН, Национальная Суперкомпьютерная Технологическая Платформа (НСТП), Евразийская Суперкомпьютерная Технологическая Платформа (ЕСТП). - [Б/м], 2019. - С. 10 с. – URL : http://2019.nscf.ru/TesisAll/06_Prikladnoe_PO/161_EgynovVA.pdf.

Подписано в печать ___.___.2025г. Заказ №___.
Тираж ___ экз. Усл. печ. л. 2,0
Формат 60 х 84 1/16. Бумага офсетная. Печать офсетная.
Издательство Волгоградского государственного технического университета.
400005, г. Волгоград, пр. им. В.И. Ленина, 28, корп. №7