

Разреженная линейная алгебра: от аппаратной поддержки до применений

Семён Григорьев

25 февраля 2021 г.

1 Сведения о проекте

1.1 Название проекта

ru

!!!

или

!!!

или

!!!!

en

1.2 Приоритетное направление развития науки, технологий и техники в Российской Федерации, критическая технология

1.3 Направление из Стратегии научно-технологического развития Российской Федерации (утверждена Указом Президента Российской Федерации от 1 декабря 2016 г. №642 "О Стратегии научно-технологического развития Российской Федерации") (при наличии)

1.4 Ключевые слова (приводится не более 15 терминов)

ru

Высокопроизводительные вычисления, линейная алгебра, массово-параллельные архитектуры, разреженные структуры данных, графовые базы данных, алгоритмы анализа графов,

статический анализ кода, параллельные алгоритмы, суперкомпиляция, смешанные вычисления, специализация, язык запросов к графам.

en

High-performance computing, linear algebra, massively-parallel hardware, sparse data structures, graph database, graph analysis algorithms, Static code analysis, parallel algorithms, supercompilation, mixed computations, partial evaluation, specialization, graph query languages.

1.5 Аннотация проекта

ru

Эффективная обработка больших объёмов данных — актуальная прикладная область, требующая качественных теоретических результатов для решения возникающих прикладных задач. Одной из активно изучаемых в последнее время моделей для представления обрабатываемых данных является граф. На практике такая модель применяется при работе с различными сетями (социальные сети, транспортные сети), при анализе и верификации программных и аппаратных комплексов (графы вызовов, переходов и т.д.), а в общем случае является основой для графовых баз данных. Иными словами, обработка граф-структурированных данных является активно развивающейся областью.

Одна из задач при обработке данных — поиск и анализ связей между сущностями (или же установление факта отсутствия специфических связей). В случае граф-структурированных данных данная задача формулируется в терминах поиска путей между вершинами или проверки их отсутствия. При этом содержательные задачи используют дополнительные, не тривиальные, ограничения на пути. В качестве классического примера дополнительных ограничений можно рассмотреть поиск простых путей и поиск кратчайших путей в графе.

Одним из способов задать ограничение на путь в размеченном графе (то есть в графе, рёбра которого несут некоторую нагрузку в виде метки или веса) основан на использовании формальных языков. В данном случае рассматриваются слова, полученные конкатенацией меток рёбер пути, и задаётся язык, которому должны принадлежать такие слова. Иными словами, возникает следующая задача: найти такие пути в графе, что слова, задаваемые ими, принадлежат заданному языку. При этом возможны различные вариации постановки задачи (характерные для многих задач поиска путей): поиск пути между двумя заданными вершинами; поиск всех путей в графе, удовлетворяющих заданному ограничению; проверка достижимости (а не поиск непосредственно пути) и т.д. В зависимости от конкретной решаемой задачи необходимо применять различные алгоритмы для достижения лучшей эффективности.

Так как ограничения формулируются в терминах языков, естественным является привлечение результатов теории формальных языков. С одной стороны, возникают фундаментальные вопросы о разрешимости задачи: при использовании каких классов языков в качестве ограничений задача поиска путей разрешима. С другой стороны, оказывается возможным использовать алгоритмы синтаксического анализа для решения задачи, однако они требуют

модификации, и исследование их теоретических свойств, например, временной и пространственной сложности, оказывается нетривиальной задачей. Важно, что ответы на эти вопросы связаны не только со свойствами используемых языков, но и со свойствами обрабатываемых графов, что приводит к тесному соприкосновению двух областей науки: теории графов и теории формальных языков. Несмотря на то, что задача поиска путей с ограничениями в терминах формальных языков изучается с начала 1990-х (Томас Репс и Михалис Яннакакис), многие вопросы остаются открытыми. Например, до сих пор не решён вопрос о существовании субкубического алгоритма для поиска путей с контекстно-свободными ограничениями. А конкретные алгоритмы решения задач стали разрабатываться и изучаться совсем недавно, когда возрос интерес к графовым базам данных.

С прикладной точки зрения, важно получение эффективных с вычислительной точки зрения алгоритмов для обработки практически важных сценариев. Так как графы, возникающие в прикладных задачах, имеют большой размер в терминах количества вершин и рёбер, то естественным путём является разработка параллельных и распределённых алгоритмов их обработки, в том числе алгоритмов, использующих массово-параллельные архитектуры, такие как GPGPU. Данное направление активно развивается в области обработки графов, однако слабо проработано в контексте обсуждаемой задачи.

Если рассматривать задачу поиска путей в контексте графовых баз данных, то необходимо предоставить удобные средства описания запросов, позволяющие формулировать ограничения в терминах формальных языков. Одним из наиболее естественных способов задавать такие ограничения в прикладных задачах являются парсер-комбинаторы. Традиционно парсер-комбинаторы используются для задания языка и, одновременно, синтаксического анализатора для него, путём комбинирования функций, реализующих более простые парсеры. Парсер-комбинаторы обеспечивают при этом "бесшовную" интеграцию с основным языком программирования (нет отдельной процедуры встраивания специализированного языка в язык общего назначения), высокий уровень абстракции за счёт возможности использовать функции высших порядков, надёжность и безопасность за счёт полной интеграции с системой вывода типов используемого языка. Такой подход хорошо зарекомендовал себя при анализе языков программирования, однако его применимость для анализа графов исследована слабо.

Также, в контексте выполнения запросов к графовым базам данных, необходимо разработать методы оптимизации как самих запросов, так и процедур их исполнения. Здесь перспективным подходом является применение смешанных вычислений, в частности, специализации. Хотя в области реляционных баз данных такой подход показал себя состоятельным (например, работы Евгения Шарыгина и соавторов), в контексте графовых баз данных данные техники не применялись.

Проект посвящён разработке и реализации алгоритмов для поиска путей с ограничениями в терминах формальных языков, а также вопросам создания средств задания таких ограничений и методам оптимизации соответствующих запросов к графовым базам данных. При разработке алгоритмов будут использоваться методы теории формальных языков и теории графов для поиска классов графов и языков, для которых, во-первых, принципиально возможно построение алгоритмов решения задач поиска путей с ограничениями в терминах формальных языков, во-вторых, возможно построение асимптотически эффективных алгоритмов. Для разработки эффективных с практической точки зрения алгоритмов будут

использоваться методы построения параллельных алгоритмов, в том числе, алгоритмов для массово-параллельных архитектур. Исследование способов задания ограничений потребует использования знаний из области разработки языков программирования. При разработке методов оптимизации запросов будут использоваться техники смешанных вычислений.

Коллектив исполнителей включает специалистов по теории формальных языков, теории графов, построению компиляторов, методам оптимизации программ, и разработке языков программирования. Это позволит организовать плодотворное сотрудничество и обеспечить комплексный подход к решению задач, а также привлечь талантливых студентов к изучению соответствующих областей науки и работе над проектом.

en

1.6 Ожидаемые результаты и их значимость

ru

Проект направлен на изучение задачи о поиске путей с ограничениями в терминах формальных языков с целью получения эффективного с прикладной точки зрения решения для неё. Ожидаются как теоретические результаты на стыке теории формальных языков и теории графов и в области построения параллельных алгоритмов, так и результаты в области разработки языков и методов оптимизации программного обеспечения.

В частности, ставится задача построить более детальную классификацию задач поиска путей с контекстно-свободными ограничениями как с точки зрения подклассов языков, так и с точки зрения типов графов. Основная цель здесь — ответить на вопрос о существовании субкубического алгоритма для задачи в общем случае. Данный вопрос открыт уже длительное время, так что полностью ответить на него вряд ли удастся, но ценными будут и частичные ответы в терминах подклассов задачи, для которых такой алгоритм точно существует.

В области построения параллельных алгоритмов планируется получение новых алгоритмов для решения задачи поиска путей с контекстно-свободными ограничениями для массово-параллельных и распределённых систем. Будут изучены теоретические свойства предложенных алгоритмов, в частности, получены асимптотические оценки временной и пространственной сложности. Также будет исследованы возможности расширения построенных алгоритмов для других классов языков.

При разработке прикладных способов и средств задания ограничений в терминах языков будут исследованы подходы на основе парсер-комбинаторов. Планируется, что будут сформулированы границы применимости такого подхода, а также изучены его слабые и сильные стороны в контексте прикладных задач, такие как типобезопасность, возможность вычисления дополнительных семантических функций. Несмотря на то, что применение парсер-комбинаторов для анализа языков программирования изучено достаточно хорошо, обобщение этого подхода на графы нетривиально и ожидаются новые результаты. Парсер-комбинаторы предоставляют не только механизм для решения задачи поиска путей с ограничениями, но и формализм для описания запросов. Использование такого формализма упростит использование технологии конечными пользователями, а также предоставит бо-

лее прозрачную интеграцию в логику разрабатываемой программы. Планируется разработка удобного формализма спецификации запросов. Помимо того, парсер-комбинаторы позволяют вычисление пользовательской семантики, при помощи чего можно выразить фильтрацию, агрегацию, счетчики и прочие виды обработки результата запроса. В рамках работы будет изучено, для каких классов входных графов можно точно вычислить пользовательскую семантику, а для каких только приближённо. Некоторые языки, не являющиеся контекстно-свободными, можно анализировать при помощи парсер-комбинаторов. Будет изучен вопрос использования более, чем контекстно-свободных ограничений для поиска путей в графах.

В области оптимизации запросов и процедур их исполнения планируется разработать решение для специализации алгоритмов выполнения запросов к графовым базам данных во время выполнения. Вероятно, при этом будут разработаны новые алгоритмы специализации.

en

1.7 +В состав научного коллектива будут входить

- 8 исполнителей проекта (включая руководителя)
- в том числе 8 исполнителей в возрасте до 39 лет включительно,
- из них: 4 очных аспирантов, адъюнктов, интернов, ординаторов, студентов.

1.8 +Планируемый состав научного коллектива с указанием фамилий, имен, отчеств (при наличии) членов коллектива, их возраста на момент подачи заявки, ученых степеней, должностей и основных мест работы, формы отношений с организацией (трудовой договор, гражданско-правовой договор) в период реализации проекта.

- Семён Вячеславович Григорьев, 31 год, к.ф.-м.н., доцент СПбГУ, трудовой договор.
- Даниил Андреевич Березун, 27 лет, к.ф.-м.н., научный сотрудник ООО "ИнтеллиДжей Лабс приглашённый лектор в НИУ ВШЭ, гпд
- Антон Подкопаев
- Тимофей Брыксин
- Рустам Шухратуллович Азимов, 24 года, магистр (математическое обеспечение и администрирование информационных систем) ,научный сотрудник ООО "ИнтеллиДжей Лабс трудовой договор
- Егор Орачев

- Алексей Тюрин
- Арсений Константинович Терехов, 21 год, студент СПбГУ, гпд

+Соответствие профессионального уровня членов научного коллектива задачам проекта

гц

Руководитель, Семён Вячеславович Григорьев, является доцентом кафедры информатики СПбГУ и кандидатом физико-математических наук. Опыт руководства исследовательскими работами и преподавания составляет 7 лет. За это время под его руководством защищено 8 магистерских диссертаций, 15 выпускных квалификационных работ бакалавра, 2 дипломных работы специалиста, больше 15 курсовых работ. В настоящее время под его руководством работают два аспиранта. За время преподавательской деятельности занимался подготовкой и чтением курсов по теории графов, алгоритмам анализа графов, теории формальных языков, алгоритмам и структурам данных. Имеет опыт руководства грантами (РФФИ 19-37-90101; программа УМНИК, 162ГУ1/2013 и 5609ГУ1/2014) исследовательскими группами и отдельными исследовательскими работами. Также имеет опыт исполнения грантов (РФФИ 15-01-05431, РФФИ 18-01-00380, РНФ 18-11-00100). Область научных интересов включает теорию формальных языков, теорию графов, алгоритмы синтаксического анализа, разработку параллельных алгоритмов, аппаратные ускорители параллельных вычислений.

Даниил Андреевич Березун является кандидатом физико-математических наук, преподаёт на кафедре прикладной математики и информатики НИУ ВШЭ в Санкт-Петербурге. Опыт руководства исследовательскими работами и преподавательской деятельности составляет более 5 лет. За это время под его руководством были защищены 3 выпускных квалификационных работы бакалавра, более 6 курсовых работ. За время преподавательской деятельности занимался подготовкой и чтением курсов по компиляции, разработке языковых процессоров, метавычислениям и семантикам языков программирования. В настоящее время под его руководством работают 2 магистранта. Имеет опыт исполнения грантов (РФФИ 18-01-00380). Область научных интересов включает анализ, разработку и реализацию языков программирования, метапрограммирование и метавычисления, математическую логику, семантику языков программирования, автоматическую генерацию программ, основанную на семантике, блокчейн и распределённые технологии.

Подкопаев Антон

Брыксин Тимофей

Рустам Шухратуллович Азимов является аспирантом математико-механического факультета СПбГУ по направлению информатика. Защитил магистерскую диссертацию на тему "Синтаксический анализ графов через умножение матриц". Имеет публикации по теме проекта ("Context-Free Path Querying by Matrix Multiplication "Синтаксический анализ графов с использованием конъюнктивных грамматик "Синтаксический анализ графов и задача генерации строк с ограничениями"). Имеет опыт исполнения грантов (РНФ 18-11-00100 и РФФИ 19-37-90101). Область научных интересов: теория формальных языков, запросы к графам, языки запросов, поиск путей в графах, матричные операции, параллельные алгоритмы.

Терехов Арсений является студентом 4го курса СПбГУ по направлению "Математическое обеспечение и администрирование информационных систем" а так же студентом 3го курса Computer Science Center. Прошёл летние стажировки в компаниях Яндекс и JetBrains. Принимал участие в двух проектах под руководством работников компании JetBrains. Его область научных интересов включает формальные языки и графовые базы данных.

Тюрин Алексей

Орачев Егор

en

The lead of the group, Semyon V. Grigorev, is an associate professor of the faculty of Mathematics and Mechanics of Saint-Petersburg State University and has a Ph.D. in mathematics and physics. He has 7 years of experience in teaching and being a leader and a manager of research projects. He has supervised 8 master dissertations, 15 graduation theses of bachelors, 2 graduation theses of specialists, and more than 15 course works. Two Ph.D. students are being supervised by him now. The following courses were prepared and taught: graph theory, formal language theory, algorithms and data structures. Semyon has experience in being a leader of both grants (RFBR 19-37-90101; FASIE, 162ГУ1/2013 and 5609ГУ1/2014), and research groups and projects. Also, he has participated in grants (RFBR 15-01-05431, RFBR 18-01-00380, RSF 18-11-00100). Research interests include formal language theory, graph theory, parsing algorithms, parallel algorithms, high-performance computing, hardware accelerators.

Daniil A. Berezun has a Ph.D. in mathematics and physics, and is a lecturer at the Applied Mathematics and Informatics chair of NRU HSE in St. Petersburg. He has supervised 3 graduation theses of bachelor and more than 6 course works. Two master students are being supervised by him now. The following courses were prepared and taught: compiler techniques, language processors development, programming languages semantics, metacomputations. Daniil has participated in the grant RFBR 18-01-00380. Research interests include analysis, design, and implementation of programming languages, programming languages semantics, metaprogramming and metacomputations, semantic-based automated program generation, blockchain, and distributed systems.

Rustam Sh. Azimov is a Ph.D. student at the faculty of Mathematics and Mechanics at Saint-Petersburg State University. He has a masters degree, his masters thesis is "Graph parsing by matrix multiplication". Rustam has publications which are related to this project ("Context-Free Path Querying by Matrix Multiplication" "Graph parsing by using conjunctive grammars" "Graph parsing and constrained string generation problem"). He has participated in grants (RSF 18-11-00100 и RFBR 19-37-90101). Research interests include formal language theory, graph querying, query languages, linear algebra, parallel algorithms.

Arseniy K. Terekhov is a 4th-year student at the faculty of Mathematics and Mechanics at Saint-Petersburg State University, specialization is "Information System Administration" and is a 3rd-year student of Computer Science Center. Arseniy was an intern at Yandex and JetBrains software development company. He worked on two research projects led by employers of JetBrains. Research interests include formal languages and graph databases.

Anton Podkopaev

Alexei Turin

Timofey Briksin

Egor Orachov

1.9 +Планируемый объем финансирования проекта Фондом по годам (указывается в тыс. рублей)

2021 г. 6000 тыс. рублей, 2022 г. 6000 введите планируемый объем финансирования в 2022 г. тыс. рублей, 2023 г. 6000 введите планируемый объем финансирования в 2023 г. тыс. рублей.

1.10 Научный коллектив по результатам проекта в ходе его реализации предполагает опубликовать в рецензируемых российских и зарубежных научных изданиях не менее

10 публикаций

из них 8 в изданиях, индексируемых в базах данных «Сеть науки» (Web of Science Core Collection) или «Скопус» (Scopus).

Информация о научных изданиях, в которых планируется опубликовать результаты проекта, в том числе следует указать в каких базах индексируются данные издания - «Сеть науки» (Web of Science Core Collection), «Скопус» (Scopus), РИНЦ, иные базы, а также указать тип публикации - статья, обзор, тезисы, монография, иной тип

- Proceedings of Joint International Workshop on Graph Data Management Experiences & Systems (Grades) and Network Data Analytics (Nda), издатель ACM, Scopus, статья
- Proceedings of International Conference on Extending Database Technology (EDBT), издатель OpenProceedings.org, Scopus, статья
- Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation, издатель ACM, Scopus, статья
- IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, издатель IEEE, Scopus, статья
- Lecture Notes in Computer Science, издатель Springer US, Scopus, Web of Science, статья

Иные способы обнародования результатов выполнения проекта

- Участие в постерных сессиях при конференциях SIGMOD, SPLASH, ICFP
- Проведение открытых лекций
- Доклады на научных семинарах

1.11 Число публикаций членов научного коллектива, опубликованных в период с 1 января 2015 года до даты подачи заявки

25, из них 10 – опубликованы в изданиях, индексируемых в Web of Science Core Collection или в Scopus.

1.12 Планируемое участие научного коллектива в международных коллаборациях (проектах) (при наличии)

Антон? Тимофей?

Руководитель проекта подтверждает, что

- все члены научного коллектива (в том числе руководитель проекта) удовлетворяют пунктам 6, 7, 13 конкурсной документации;
- на весь период реализации проекта он будет состоять в трудовых отношениях с организацией;
- при обнародовании результатов любой научной работы, выполненной в рамках поддержанного Фондом проекта, он и его научный коллектив будут указывать на получение финансовой поддержки от Фонда и организацию, а также согласны с опубликованием Фондом аннотации и ожидаемых результатов поддержанного проекта, соответствующих отчетов о выполнении проекта, в том числе в информационно-телекоммуникационной сети «Интернет»;
- помимо гранта Фонда проект не будет иметь других источников финансирования в течение всего периода практической реализации проекта с использованием гранта Фонда;
- проект не является аналогичным по содержанию проекту, одновременно поданному на конкурсы научных фондов и иных организаций;
- проект не содержит сведений, составляющих государственную тайну или относимых к охраняемой в соответствии с законодательством Российской Федерации иной информации ограниченного доступа;
- доля членов научного коллектива в возрасте до 39 лет включительно в общей численности членов научного коллектива будет составлять не менее 50 процентов в течение всего периода практической реализации проекта;
- в установленные сроки будут представляться в Фонд ежегодные отчеты о выполнении проекта и о целевом использовании средств гранта.

2 Содержание проекта

2.1 +Научная проблема, на решение которой направлен проект

ru

Проект направлен на изучение и разработку методов и средств, позволяющих использовать операции и примитивы линейной алгебры для получения высокопроизводительных решений в широком спектре прикладных областей, включающем (но не ограниченном) статический анализ кода, графовые базы данных, анализ социальных сетей.

Хотя прикладное значение линейной алгебры хорошо известно, наиболее широкое распространение на практике она получила в виде математических библиотек линейной алгебры (например, различные реализации BALS, Basic Linear Algebra Subprograms). При этом, применение алгоритмов, основанных на линейной алгебре, в различных прикладных областях хотя и является традиционным (например, операции над матрицами смежности при анализе графов), до недавнего времени не существовало единого взгляда на такой подход к решению прикладных задач. Появление же стандарта GraphBLAS (Aydin Buluç, Timothy Mattson, Scott McMillan, Jos'e Moreira, Carl Yang, The GraphBLAS C API Specification, version 1.3.0, 2019), с одной стороны, привело к общему базису использования примитивов и операций линейной алгебры в широком спектре прикладных задач, с другой же, поставило ряд вопросов, ответы на которые активно ищутся мировым научным сообществом в настоящее время. При этом вопросы затрагивают все уровни, начиная от аппаратного, без работы над которым не возможно обеспечить максимальную эффективность решений, через системный, включающий разработку библиотек и средств разработки, позволяющих, с одной стороны, минимизировать затраты на разработку прикладных решений основанных на линейной алгебре, с другой, эффективно использующих имеющиеся аппаратные средства, до прикладного, включающего изучение областей, в которых перспективно применение решений на основе операций и примитивов линейной алгебры, разработку инструментов и приложений для конечных пользователей, основанных на линейной алгебре.

На аппаратном и системном уровне существенная часть вопросов связана со следующими ключевыми особенностями GraphBLAS. Во-первых, данный стандарт рассматривает прежде всего разреженные структуры данных (разреженные матрицы, вектора), что связано с разреженностью большинства прикладных объектов. Это существенно снижает эффективность традиционных аппаратных и программных средств параллельных вычислений (такие как использование векторизации, SIMD, GPGPU) из-за нерегулярного доступа к памяти, вызванного особенностью форматов представления данных. Что, в свою очередь, стимулирует поиск новых аппаратных архитектур и моделей и средств программирования для них и для уже существующих платформ. Во-вторых, постулируется необходимость реализации абстрактных примитивов и операций, параметризуемых теми объектами, как моноид или полукольцо, что существенно отличает его от принятого подхода в современных библиотеках линейной алгебры, поддерживающих, в основном, числа с плавающей точкой (с одинарной или двойной точностью) и соответствующие операции. В-третьих, набор примитивов и операций разрабатывается исходя из того, что они должны стать строительными блоками для прикладных алгоритмов, что выдвигает более нетривиальные требования как к составу самого набора, так и к возможности легко строить композиции операций, входящих в него. Последние два пунк-

та вынуждают искать новые подходы к разработке библиотек реализаций операций линейной алгебры и алгоритмов на их основе, так как необходимо совместить разработку высокопроизводительных решений, традиционно ведущуюся на языках низкого уровня, и создание удобного уровня абстракций, возможное, в основном, с использованием соответствующих языков высокого уровня. Это, в свою очередь, приводит к необходимости разрабатывать новые языки программирования и техники оптимизаций программ, написанных на них.

Сложность при работе в данном направлении заключается в том, что традиционно для разработки высокопроизводительных решений использовались императивные (часто C-подобные) языки, однако многие востребованные оптимизации, такие как устранение промежуточных структур данных (в частности дефорестация), частичное применение к известным данным (специализация), реализуемы в языках, тяготеющих к функциональной парадигме. Стоит отметить, что указанные выше оптимизации могут рассматриваться как частные случаи суперкомпиляции, предложенной и активно изучавшейся В.Ф.Турчиным и А.П.Ершовым ещё в 70-е годы. И хотя работы по созданию языка для высокопроизводительных вычислений, поддерживающих подобные оптимизации активно ведутся в настоящее время (например, язык Futhark, поддерживающий kernel fusion, частный случай дефорестации: Троэлс Хенриксен, "Futhark: Purely Functional GPU-Programming with Nested Parallelism and In-Place Array Updates 2017) и обсуждаются в сообществе GraphBLAS (Карл Янг "GraphBLAST: A High-Performance Linear Algebra-based Graph Framework on the GPU 2020), вопрос о границах применимости таких оптимизаций и о применимости суперкомпиляции для их выполнения всё ещё остаётся открытым.

Вместе с этим, прикладной уровень, где ведётся поиск новых областей применения и способов использования линейной алгебры, нуждается в усиленном изучении и структуризации. Так, например, хотя использование операций линейной алгебры для выполнения запросов к графовым базам данных уже подтвердило свою практическую ценность, например, в графовой базе данных RedisGraph (Pieter Cailliau, Tim Davis et al, RedisGraph GraphBLAS Enabled Graph Database, 2019), практически полностью основанной на представлении графов в виде матриц и использующей операции над ними для выполнения пользовательских запросов, всё ещё не существует формального описания процедуры трансляции прикладных языков запросов в операции линейной алгебры. Хотя и предпринимаются попытки формально описать семантику некоторых языков запросов к графовым базам данных, например, в работах Надима Франсиса "Formal Semantics of the Language Cypher" и Джозефа Мартона "Formalising openCypher Graph Queries in Relational Algebra". Однако, в данных работах не используются системы автоматической проверки доказательств, хотя использование таких систем практически стало стандартом "де факто" при описании формальных свойств языков и связанных с ними инфраструктур (трансляторов, оптимизаторов, компиляторов), ввиду их сложности и невозможности вручную проверить корректность описания и доказательства свойств.

Другой показательный пример — статический анализ кода, где большой объём обрабатываемых данных делает востребованными параллельные (например, работы Хайбо Ю "Parallelizing Flow-Sensitive Demand-Driven Points-to Analysis 2020 и Поуни Гы "Towards Efficient Large-Scale Interprocedural Program Static Analysis on Distributed Data-Parallel Computation" 2021) и инкрементальные (например, работа Йенса Ван дер Пласа "Incremental Flow Analysis through Computational Dependency Reification 2020) алгоритмы (дабы избежать повторных вычислений при незначительных изменениях кода), а многие ключевые задачи могут быть сфор-

мулированы в терминах достижимости с особого вида ограничениями в графах. При этом, применимость алгоритмов для решения таких задач, построенных на основе линейной алгебры и позволяющих использовать современные программные и аппаратные средства для параллельной и инкрементальной обработки данных, в контексте статического анализа всё ещё остаётся мало изученной ввиду того, что они были предложены сравнительно недавно (работы Рустама Азимова "Context-free path querying by matrix multiplication" и Егора Орачева "Context-Free Path Querying by Kronecker Product").

en

2.2 Научная значимость и актуальность решения обозначенной проблемы

ru

Создание програмно-аппаратного стека для высокопроизводительных решений на основе линейной алгебры !!!

Необходимость подобных оптимизаций подробно обсуждается.

Детальный обзор актуальных решений приведён в !!!!

Формализация свойств и доказательство корректности преобразований необходимо как для !!!! В качестве примера можно привести проект SQLCert (<http://datacert.lri.fr/sqlcert/>), часть проекта DataCert (<http://datacert.lri.fr/>), поддерживаемого Agence Nationale de la Recherche (Франция), направленный на создание сертифицированной инфраструктуры исполнения SQL-запросов с использованием Coq, включающей и формальное описание языка запросов SQL. Необходимо создание аналогичного проекта для анализа граф-структурированных данных (в частности, графовых баз данных), особенно с применением операций линейной алгебры, так как это один из перспективных путей к высокопроизводительной обработке больших данных. Разработка систем, дающих гарантии целостности данных, безопасности, корректности обработки и т.д.

Понимание границ применимости в статическом анализе !!!

Оценка возможности использовать для машинного обучения необходима для создания интеллектуальных инструментов анализа большого объёма кода.

=====

Знание границ разрешимости задачи необходимо для разработки языков запросов и для оценки разрешимости прикладных задач, сводимых к данной. При этом, с практической точки зрения, могут оказаться содержательными ситуации, когда задача в общем случае не разрешима, но можно найти достаточно точные приближённые решения. Так, для статического анализа применимым оказывается приближение сверху, так как в большинстве случаев ожидаемый ответ пуст, что означает отсутствие нежелательных поведений анализируемой программы. А значит, если аппроксимация сверху пуста, то и точное решение пусто.

При этом важно, чтобы приближение как можно меньше отличалось от точного решения, так как в противном случае будет большое количество ложных срабатываний — ситуаций, когда найденное нежелательное поведение на самом деле не возможно. Примером такого подхода может служить работа Ц. Чжана, в которой для статического анализа кода применялись ограничения в виде линейных конъюнктивных языков (Qirun Zhang and Zhendong Su, Context-sensitive data-dependence analysis via linear conjunctive language reachability, 2017). В такой постановке задача неразрешима, однако показано, что можно эффективно искать содержательное с практической точки зрения приближенное решение.

Знание теоретических свойств алгоритмов важно как само по себе, так и для того, чтобы создавать эффективные на практике решения. Стоит отметить, что, несмотря на то, что данная область изучается уже длительное время, совсем недавно были получены новые результаты. Так, в 2017 году Ф. Брэдфорд предъявил субкубический алгоритм для задачи достижимости в случае, когда ограничения заданы языком Дика на одном типе скобок (Phillip G. Bradford, Efficient Exact Paths For Dyck and semi-Dyck Labeled Path Reachability). Предложенное решение не обобщается на произвольные контекстно-свободные ограничения и требуется дальнейшая работа в данном направлении. В 2017 году К. Чаттерджи предъявил оптимальный алгоритм проверки достижимости для специального вида графов (двунаправленные графы) в случае, когда ограничения сформулированы в виде произвольного языка Дика (Krishnendu Chatterjee, Optimal Dyck reachability for data-dependence and alias analysis). Также К. Чаттерджи показал, что предложенный алгоритм может эффективно применяться на практике для решения задач статического анализа кода.

Поиск эффективных с вычислительной точки зрения алгоритмов, в том числе алгоритмов для массово-параллельных и распределённых систем, с одной стороны важен для более глубокого понимания теоретических свойств алгоритмов и развития теории, связанной с параллельными и распределёнными системами, а с другой — для создания эффективных решения для прикладных задач, например, графовых баз данных, которые становятся всё более популярными. Как уже было сказано, поиск эффективных алгоритмов даже для хорошо изученных классов задач является актуальной на сегодняшний день проблемой (например, работы Jochem Kuijpers и Maurizio Nolé).

Исследования в области способов задания ограничений связаны с разработкой языка запросов, что является актуальной задачей. С одной стороны, языки запросов к графовым базам данных только развиваются и многие даже базовые вопросы, связанные с синтаксисом и семантикой таких языков, требуют изучения. С другой стороны, существует ряд общих вопросов, связанных с интеграцией предметно-ориентированных языков в языки общего назначения. Например, вопросы о "бесшовной" интеграции, о типовой безопасности, о различных проверках времени компиляции. Для решения этих проблем регулярно предлагаются различные подходы: интегрированный язык запросов (LINQ), различного рода комбинаторы, средства "межъязыкового" вывода типов.

Метавычисления, специализация и смешанные вычисления изучаются давно (работы N.D.Jones, В.Ф.Турчина, А.П.Ершова в 70–90-е года, а также работы их учеников, последователей и соавторов), но до сих пор в этой области много открытых вопросов как в теории, так и относительно применимости в прикладных задачах. Так, только в 2018 году специализация позволила существенно ускорить выполнение запросов в реляционной СУБД PostgreSQL (Sharygin

Eugene et.al. 2018. Runtime Specialization of PostgreSQL Query Executor), а в 2019 было показано, что с помощью суперкомпиляции возможно построить процедуру выполнения SQL-запросов, превосходящую по производительности многие аналоги (Tiark Rompf, Nada Amin. 2019. A SQL to C compiler in 500 lines of code). Производительность процедуры выполнения запросов в графовых базах данных важна с прикладной точки зрения, однако применимость данных подходов для ускорения исполнения запросов в графовых базах данных не изучалась. Вместе с тем, исследование данной области может привести к новым теоретическим задачам в области смешанных вычислений.

en

Decidability of formal language constrained path problem is important for the development of graph querying languages and also in determining the decidability of the applications which can be reduced to this problem. Note that in practical applications, when the problem is undecidable, it may be important to provide an appropriate approximation, if possible. For example in static code analysis we expect the result to be empty, since the program is expected to not contain undesirable behaviours. Thus, if the over-approximation of the result turns out to be empty, then the precise solution is empty too, which renders over-approximating the result a good idea in this application area. But it would be better to provide an approximation which is as precise as possible, because otherwise users will face too many false-positives (undesirable behavior reported when in reality program is unable to have it) and will not be able to use the technology for their benefit. The example of such an approach is a work of Q. Zhang, who uses linear conjunctive languages for static code analysis (Qirun Zhang and Zhendong Su, Context-sensitive data-dependence analysis via linear conjunctive language reachability, 2017). Linear conjunctive constrained reachability is undecidable, but it is shown that it is possible to provide a reasonable practical approximated solutions.

Theoretical properties of algorithms are important both as a self-contained theoretical result and as a means to create efficient practical solutions. Note that despite long research history, some new results were provided in the last few years. For example, in 2019, Ph. Bradford provides a subcubic algorithm for the 1-Dyck reachability problem (Phillip G. Bradford, Efficient Exact Paths For Dyck and semi-Dyck Labeled Path Reachability). The proposed solution cannot be generalized to arbitrary context-free languages, so more research in this direction is required. In 2017, K. Chatterjee provides an optimal algorithm for (arbitrary) Dyck reachability in a specific type of graph—bidirected graph—(Krishnendu Chatterjee, Optimal Dyck reachability for data-dependence and alias analysis). He also shows that the algorithm can be efficiently applied for static code analysis.

The development of computationally efficient algorithms, including algorithms for massively-parallel and distributed systems, is important for the investigation of the theoretical properties of algorithms and the advancing of parallel and distributed system theory. It also enables the creation of efficient applied solutions, for example for graph databases, which are becoming more popular in the recent years. As we note above, efficient algorithms development is a significant problem even for well-investigated classes of formal language constrained path problem (for example, see works of Jochem Kuijpers and Maurizio Nolé).

Query language development is another problem worth attention. To answer what makes a good query language, we need to investigate ways to specify constraints. Graph database query languages

are at the early stage, so there is a lot of open questions about their syntax and semantics. There is also a number of open questions about the integration of domain-specific languages into general-purpose programming languages. They deal with transparent integration, type safety, compile-time checking of correctness. Several solutions have been proposed to solve these problems (language integrated query, combinators, interlanguage type inference, etc.) and new solutions are still under development.

Partial evaluation and supercompilation have been researched since 1970s (N.D. Jones, V.F. Turchin, A.P. Ershov and their followers), but there are still many open questions both theoretical and about the applicability of these methods to real-world problems. For example, only in 2018 performance SQL query execution procedure in PostgreSQL DBMS was significantly improved by means of specialization (Sharygin Eugene et.al. 2018. Runtime Specialization of PostgreSQL Query Executor). In 2019 it was shown that it is possible to create a SQL query execution procedure which outperforms competitors by means of supercompilation (Tiark Rumpf, Nada Amin. 2019. A SQL to C compiler in 500 lines of code). Performance of query execution in graph databases is crucial for applications, but there are no results on the application of partial evaluation, supercompilation and similar to graph databases. At the same time, investigating this area can lead to new theoretical problems and tasks in the area of mixed computations and partial evaluation.

2.3 +Конкретная задача (задачи) в рамках проблемы, на решение которой направлен проект, ее масштаб и комплексность

ru

В рамках улучшения программно-аппаратной поддержки примитивов и операций разреженной линейной алгебры, предусматриваемых стандартом GraphBLAS, ставится две задачи. Первая — разработка новых подходов и инструментальных средств для создания высокопроизводительных библиотек разреженной линейной алгебры для современного аппаратного обеспечения с использованием существующих языков программирования. Здесь предполагается, с одной стороны, поиск, разработка, реализация и экспериментальное исследование алгоритмов работы с разреженными матрицами, работающих на таких аппаратных ускорителях, как графические процессоры общего назначения (GPGPU) и ПЛИС (FPGA). С другой стороны, планируется разработать подход и соответствующие инструментальные средства разработки программного обеспечения, позволяющий прозрачным для дальнейшего использования образом объединить преимущества языков высокого уровня, такие как удобные абстракции для распределённого, параллельного и асинхронного программирования или богатую систему типов, дающую существенные статические гарантии корректности кода, с высокой производительностью специализированного аппаратного обеспечения.

Вторая задача — разработка нового программно-аппаратного стека, предназначенного для разработки высокопроизводительных решений на основе операций и примитивов разреженной линейной алгебры. В рамках данной задачи планируется совместная разработка (co-design) специализированного языка программирования высокого уровня (domain specific language, DSL), позволяющего описывать необходимые операции и примитивы линейной алгебры, методов и алгоритмов оптимизации программ на разработанном DSL, его компилятора, и аппаратной архитектуры, специализированной для разработанного языка и направленной

ной на повышение производительности. Вместе с этим, планируется разработка библиотеки, максимально удовлетворяющей стандарту GraphBLAS и экспериментальное исследование полученного решения на прикладных алгоритмах.

В качестве подзадачи планируется исследование таких методов оптимизации, как суперкомпиляция, смешанные вычисления, в двух направлениях. Первое: применение суперкомпиляции для оптимизации программ, написанных на разработанном DSL. Здесь планируется изучить применимость существующих техник суперкомпиляции и их эффект на производительность целевого кода. Возможно, будут разрабатываться новые методы суперкомпиляции, специализированные для разрабатываемого языка и предметной области. Второе: применение частичных вычислений для оптимизации решений во время выполнения. Здесь необходимо найти сценарии при которых специализация на данные, становящиеся известными во время выполнения, позволяет повысить производительность решения.

В рамках исследования перспективных областей применения алгоритмов, основанных на линейной алгебре, планируется работа в двух прикладных направлениях, являющихся на сегодняшний момент одними из самых активно использующих линейную алгебру. Первое направление — это графовые базы данных. И здесь планируется изучить границы применимости линейной алгебры для выполнения запросов к графовым базам данных. Для этого планируется разработать схему трансляции языка запросов Cypher в операции линейной алгебры и доказать её корректность. Для этого планируется использовать систему автоматического доказательства теорем Coq. Отдельной задачей ставится интеграция полученных результатов с международным стандартом языка запросов к графовым базам данных (GQL Standard, <https://www.gqlstandards.org/>), так как кроме схемы трансляции планируется получить формальное описание семантики языка Cypher (на основе которого разрабатывается стандарт), а данные артефакты могут быть полезны при формальном описании языка, необходимым для стандарта.

Второе направление — статический анализ программного кода. Здесь планируется изучение применимости различных алгоритмов поиска путей с ограничениями в терминах формальных языков к статическому анализу кода и адекватность получаемых результатов для создания прикладных решений, в частности на основе статистических подходов и методов машинного обучения. Необходимо оценить эффективность алгоритмов, основанных на линейной алгебре, для анализа больших объёмов кода, в том числе, эффект от применения GPGPU для решения подобных задач. Вместе с этим, планируется разработать методы использования получаемых результатов анализов для построения прикладных решений, основанных на методах машинного обучения, что позволит улучшить интеллектуальный анализ больших объёмов кода.

en

2.4 + Научная новизна исследований, обоснование достижимости решения поставленной задачи (задач) и возможности получения запланированных результатов

ru

Рассматриваемая в проекте область активно развивается. Все поставленные задачи интересуют специалистов в соответствующих областях, что подтверждается наличием работ, опубликованных в недавнее время в рецензируемых профильных журналах и представленных на ведущих профильных конференциях, в том числе участниками проекта. Это позволяет гарантировать новизну ожидаемых результатов и их соответствие мировому уровню.

Поскольку некоторые задачи очень трудны, гарантировать их полное решение невозможно. Таковой, например, является задача разработки нового программно-аппаратного стека для высокопроизводительных решений, основанных на разреженной линейной алгебре. Однако получение даже частичных результатов или улучшение существующих (например, изучение и описание границ применимости суперкомпиляции для оптимизации подпрограмм линейной алгебры) будет существенным вкладом. Вместе с этим, в проекте предусмотрено решение ряда интересных и ожидаемо разрешимых задач.

Так, например, опыт участников в разработке и исследовании методов суперкомпиляции и смешанных вычислений (Д. Березун, А. Тюрин), а также в разрабоке, как алгоритмов на основе линейной алгебры и GraphBLAS, так и самих алгоритмов линейной алгебры, применимых для анализа графов (С.В. Григорьев, Р.Ш. Азимов, А.К. Терехов, Е. Орачев), должен позволить всесторонне подойти к изучению вопроса создания специализированного языка для описания примитивов и операций разреженной линейной алгебры и применимости метода суперкомпиляции и смешанных вычислений для оптимизации программ написанных на нём. Так как применимость суперкомпиляции в данном контексте мало изучена, получение как положительных (создание языка и эффективного суперкомпилятора для него), так и отрицательных (выводы о неприменимости тех или иных техник в рамках изучаемой задачи) результатов будет существенным вкладом.

Опыт участников в разработке алгоритмов выполнения запросов к графам, основанных на линейной алгебре, в использовании этих алгоритмов при исполнении запросов к реальным графовым базам данных (С.В. Григорьев, Р.Ш. Азимов, А.К. Терехов, Е. Орачев), в формализации различных аспектов языков программирования (в том числе семантик) и доказательстве их свойств (А.В. Подкопаев, Д.А. Березун), позволят всесторонне подойти к вопросу применимости операций линейной алгебры для выполнения запросов к графовым базам данных. Отметим, что некоторые конструкции языка Cypher будут формально изучаться с этой точки зрения впервые, однако у участников есть опыт практической реализации данных конструкций и их трансляции в операции линейной алгебры, что поможет получить здесь новые результаты. Вместе с этим, некоторые аспекты формального описания семантики языка Cypher изучены достаточно хорошо, но не полностью, и здесь планируется опираться на результаты таких исследователей, как Джозеф Мартон (József Marton) и Надим Фрэнсис (Nadime Francis).

Разработка и экспериментальное исследование параллельных инкрементальных алгоритмов статического анализа кода, основанных на достижимости с ограничениями в терминах фор-

мальных языков, активно ведётся в настоящее время, многие вопросы и прикладные задачи всё ещё не решены окончательно, поэтому анализ применимости существующих, ещё не изученных до конца, алгоритмов и разработка новых в данной области будет являться новыми результатами. При этом, участниками проекта разработан ряд алгоритмов достижимости и поиска путей с ограничениями в терминах формальных языков, позволяющих использовать современные параллельные архитектуры (многоядерные процессоры и GPGPU) и показавших свою высокую производительность и потенциал к инкрементальной обработке данных. Это позволит провести всестороннее исследование применимости данных алгоритмов в контексте статического анализа кода, и при необходимости внести в них модификации, продиктованные спецификой предметной области.

Вместе с этим, опыт участников в разработке прикладных решений анализа программного кода, в том числе, основанных на статистических методах и методах машинного обучения (Т. Брыксин), должен позволить детально изучить возможность использования результатов рассматриваемых алгоритмов для улучшения прикладных решений. Данное направление (использование результатов межпроцедурного анализа кода, представленных в матричном виде, для улучшения задач обработки кода методами машинного обучения) начало развиваться сравнительно недавно, однако результаты Юлей Суй (Yulei Sui), представленные в 2020 году, показывают его перспективность. Имеющийся у участников проекта опыт должен позволить существенно продвинуться в данном направлении и получить новые результаты.

en

2.5 Современное состояние исследований по данной проблеме, основные направления исследований в мировой науке и научные конкуренты

ru

В последние годы роль линейной алгебры в различных прикладных областях существенно возрасла. В частности, был разработан стандарт GraphBLAS API (Aydin Buluç, Timothy Mattson, Scott McMillan, Jos'e Moreira, Carl Yang, The GraphBLAS C API Specification, version 1.3.0, 2019) описывающий примитивы линейной алгебры и операции над ними, необходимые для описания алгоритмов анализа графов. Необходимо отметить, что данный стандарт показал свою применимость далеко за пределами задач, связанных с анализом графов. Например, в таких областях, как машинное обучение (J. Kepner, M. Kumar, J. Moreira, P. Pattnaik, M. Serrano, and H. Tufo, Enabling massive deep neural networks with the GraphBLAS, 2017) и биоинформатика (O. Selvitopi, S. Ekanayake, G. Guidi, G. Pavlopoulos, A. Azad, and A. Buluc, Distributed many-to-many protein sequence alignment using sparse matrices, 2020). В настоящее время исследователи всего мира работают над развитием данного подхода. В частности, ищутся возможности сведения классических фундаментальных и прикладных задач к операциям линейной алгебры (например, частичное решение сведения поиска в глубину получено

только в 2019 году Daniele G. Spampinato в работе "Linear Algebraic Depth-First Search a полного решения всё ещё не существует), разрабатываются новые алгоритмы для выполнения базовых операций, исследуются форматы представления разреженных структур данных, характерных для прикладных областей, разрабатываются архитектуры аппаратных ускорителей операций линейной алгебры и принципы их построения.

В частности, предпринималось несколько попыток реализации стандарта GraphBLAS на графических ускорителях общего назначения. Наиболее успешный проект возглавляет Айдын Булуç (Aydın Buluç, GraphBLAST: A High-Performance Linear Algebra-based Graph Framework on the GPU). Кроме того, Тимоти Дэвис (Timothy A. Davis, Algorithm 1000: SuiteSparse:GraphBLAS: Graph Algorithms in the Language of Sparse Linear Algebra, 2019) также планирует использование графических ускорителей в своей реализации стандарта. Однако, в этих проектах, столкнулись с несколькими серьёзными проблемами, обсуждаемыми, в том числе, в указанной статье. Первая — необходимость специфичных оптимизаций, устраняющих промежуточные структуры данных и объединяющих код, работающий с одними и теми же данными (kernel fusion), которые трудно реализуемы для Си-подобных языков, на которых в настоящее время ведутся разработки указанных проектов. Вместе с этим, подобные оптимизации хорошо изучены для других, более высокоуровневых и дающих больше статических гарантий языков (Т. Henriksen, N. G. W. Serup, M. Elsmann, F. Henglein, and C. E. Oancea, "Futhark: Purely functional gpu-programming with nested parallelism and in-place array updates", 2017) и структур данных (О. Kiselyov, A. Biboudis, N. Palladinos, and Y. Smaragdakis, "Stream fusion, to completeness", 2017). Но результаты данных исследований показывают, что выбранные в них способы решения проблемы весьма ограничены и, в частности, не позволяют реализовывать операции разреженной линейной алгебры. С другой стороны, для функциональных языков существует более общая техника, называемая дефорестацией, позволяющая выполнять требуемые оптимизации (Р. Wadler, "Deforestation: transforming programs to eliminate trees 1990). Более общий подход к оптимизациям, включающий данную технику, называется суперкомпиляцией и хорошо изучен. Так, группы под руководством В.Ф.Турчина занимались изучением автоматического преобразования программ посредством суперкомпиляции. Создание применимых на практике решений, основанных на данных методах оптимизации программ является активно исследуемой областью. Так, например, И.Г.Ключников и С.А.Романенко в 2009 году сумели применить идеи суперкомпиляции для функций высших порядков (И.Г.Ключников. Суперкомпиляция функций высших порядков. 2010), а затем предложили многорезультатную и многоуровневую суперкомпиляцию (I.Klyuchnikov, S.A.Romanenko. Multi-result supercompilation as branching growth of the penultimate level in metasystem transitions. 2011; I.Klyuchnikov, S.A.Romanenko. Higher-level supercompilation as a metasystem transition. 2012) позволяющие ещё лучше оптимизировать программы. Однако, существенной проблемой на пути применения суперкомпиляции является то, что создание эффективного суперкомпилятора для реального языка программирования — это трудоёмкая задача и даже если она решена, всё равно эффект примерения суперкомпилятора в общем случае остаётся непредсказуемым.

Параллельно ведутся разработки специализированных архитектур процессоров для операций линейной алгебры. Стоит разделить два направления. Первое — создание процессоров для решения задач, связанных с анализом графов. Подробный анализ текущего состояния области приводится в работе Y. Horawalavithana "On the design of an efficient hardware accelerator

for large scale graph analytics” (2016). Среди работ в данном направлении выделяется работа W. S. Song, V. Gleyzer, A. Lomakin, and J. Kepner, “Novel graph processor architecture, prototype system, and results” (2016) тем, что в ней предлагается создать процессор, набор инструкций которого ориентирован на операции разреженной линейной алгебры. Однако, данная работа ограничивается операцией перемножения матриц, чего не достаточно для реализации всех необходимых функций. Второе направление связано с разработкой специализированных архитектур для определённых операций линейной алгебры (чаще всего это умножение разреженной матрицы на разреженную матрицу). В этой области активно работают несколько групп и ими опубликованы следующие работы: Z. Zhang, H. Wang, S. Han, and W. J. Dally, “Sparch: Efficient architecture for sparse matrix multiplication”(2020), M. Soltaniyeh, R. P. Martin, and S. Nagarakatte, “Synergistic cpu-fpga acceleration of sparse linear algebra” (2020), S. Pal, J. Beaumont, D. Park, A. Amarnath, S. Feng, C. Chakrabarti, H. Kim, D. Blaauw, T. Mudge, and R. Dreslinski, “Outerspace: An outer product based sparse matrix multiplication accelerator” (2018). Недостатком этих работ является то, что они нацелены на ускорение одной конкретной операции. Общим недостатком всех указанных работ является то, что они не включают в рассмотрение оптимизации на уровне языка и его компилятора (оптимизатора), а ведь многие оптимизации выполнимы только на этом уровне. Например, выполнение упомянутого выше “спешивание” операций с целью устранить промежуточные структуры данных (скажем, про последовательном сложении нескольких матриц) на аппаратном уровне вряд ли возможно, в то время как на уровне компилятора для этого существуют указанные ранее техники.

Учитывая то, что перспективные оптимизации лучше всего разработаны для функциональных языков программирования, необходимо обратить внимание на работы, связанные с компиляцией программ на функциональных языках в специализированные аппаратные архитектуры. С одной стороны, активно разрабатываются специализированные процессоры для функциональных языков программирования: M. Naylor and C. Runciman, “The Reduceron: Widening the Von Neumann bottleneck for graph reduction using an fpga (2008); A. Boeijink, P. K. F. Hölzenspies, and J. Kuper, “Introducing the Pilgrim: A processor for executing lazy functional languages” (2011); R. Coelho, F. Tanus, A. Moreira, and G. Nazar, “Acqua: A parallel accelerator architecture for pure functional programs” (2020). Данные проекты нацелены на создание процессора общего назначения, специализированного для функциональных языков программирования, и на текущий момент не достаточно производительны, чтобы послужить платформой для ускорения операций линейной алгебры. С другой же стороны, активно развивается направление, позволяющее компилировать код на функциональных языках в архитектуры, специализированные для данного кода. Таким образом, этот подход позволяет получать проблемно-специфичные ускорители для конкретных задач. Данный подход хорошо исследован в работах Ричарда Тоунсвенда (Richard Townsend, "Compiling Irregular Software to Specialized Hardware"(2019), "From functional programs to pipelined dataflow circuits"(2017)) под руководством Стефена Эдвардса (Stephen A. Edwards). Данный подход перспективен для решения поставленных в рамках данного проекта задач, однако в указанных исследованиях не уделяется особого внимания высокоуровневым оптимизациям исходных программ и изучению применимости данного подхода для создания ускорителей для алгоритмов, основанных на операциях линейной алгебры.

Формализацией семантик языков запросов к данным и построением сертифицированных в Coq средств анализа и исполнения запросов активно занимается группа исследователей

в рамках проекта DataCert (<http://datacert.lri.fr/>), цель которого сертифицировать и про-
 верифицировать с использованием системы Coq системы обработки данных. В рамках дан-
 ного проекта описана в Coq семантика языка SQL: Véronique Benzaken, Evelyne Contejean,
 SQLCert: Coq mechanisation of SQL's compilation: Formally reconciling SQL and (relational)
 algebra, 2017. Для языков запросов к графам подобные результаты ещё не представлены, од-
 нако, для языка Cypher описана его формальная семантика "на бумаге": József Marton, Gábor
 Szárnyas, and Dániel Varró, Formalising openCypher Graph Queries in Relational Algebra (2017);
 Nadime Francis, et al, Formal Semantics of the Language Cypher (2018). Кроме этого, активно
 развивается проект, описывающий семантику Cypher в терминах языка Datalog: Filip Murlak,
 Jan Posiadała, and Paweł Susicki, "On the semantics of Cypher's implicit group-by"(2019). Од-
 нако, данные работы не используют средств автоматической проверки доказательств (типа
 Coq), не рассматривают вопрос трансляции конструкций исходного языка в операции линей-
 ной алгебры, и не описывают некоторые важные с прикладной и содержательные с исследова-
 тельской точек зрения особенности языка, такие как недавно предложенное Тобиасом Линда-
 кером расширение, позволяющее описывать регулярные и контекстно-свободные ограниче-
 ния на пути ((Tobias Lindaaker, Path Pattern, <https://github.com/thobe/openCypher/blob/rpq/cip/1.ассер>
 02-06-Path-Patterns.adoc). Отметим, что данное расширение было недавно реализовано на
 пратике и показало свою применимость для решения прикладных задач (Arseniy Terekhov,
 et al, "Multiple-Source Context-Free Path Querying in Terms of Linear Algebra (2021)). Это де-
 лает задачу формального описания данного расширения более актуальной.

Разработка и исследование параллельных инкрементальных алгоритмов межпроцедурно-
 го статического анализа кода активно ведётся несколькими группами исследователей. За по-
 следнее время получен ряд результатов, включающий, в том числе, результаты Ю Су и коллег
 (Y. Su, D. Ye, J. Xue and X. Liao, "An Efficient GPU Implementation of Inclusion-Based Pointer
 Analysis 2016; "Parallel Pointer Analysis for Large-Scale Software 2015), показавших перспек-
 тивность использования графических ускорителей в данной области. Также, в данной обла-
 сти известны результаты Ронг Гу (R. Gu et al., "Towards Efficient Large-Scale Interprocedural
 Program Static Analysis on Distributed Data-Parallel Computation 2021), рассматривающего,
 прежде всего, распределённые алгоритмы, а также результаты Хайбо Ю (H. Yu, Q. Sun,
 K. Xiao, Y. Chen, T. Mine and J. Zhao, "Parallelizing Flow-Sensitive Demand-Driven Points-to
 Analysis 2020) и Цзишэн Чжао (Jisheng Zhao, Michael G. Burke, and Vivek Sarkar, "Parallel
 sparse flow-sensitive points-to analysis 2018). Данные работы основаны на наблюдении, что
 многие виды межпроцедурного анализа, как было показано ещё Томасом Репсом, сводимы к
 поиску путей с контекстно-свободными ограничениями в графе, и в них предлагаются раз-
 личные варианты параллельных алгоритмов для решения данной задачи. Однако, в совре-
 менных работах не рассматривается подход, основанный на линейной алгебре. Вместе с этим,
 для поиска путей с контекстно-свободными ограничениями за последнее время был предло-
 жен ряд аогоритмов, основанных на операциях линейной алгебры, и было показано, что такие
 алгоритмы позволяют использовать современные аппаратные средства для параллельных вы-
 числений, такие как многоядерные центральные процессоры и графические ускорители общего
 назначения. Сделано это было в работах Рустама Азимова (Rustam Azimov, et al, "Context-
 free path querying by matrix multiplication 2018), Терехова Арсения (Arseniy Terekhov, et al,
 "Context-Free Path Querying with Single-Path Semantics by Matrix Multiplication 2020), Еро-
 па Орачева (Egor Orachev, et al, "Context-Free Path Querying by Kronecker Product 2020).

Однако, данные работы рассматривали применение указанных алгоритмов а графовых базах данных, где аналогичная задача, поставленная Михалисом Яннакакисом, изучалась во многом независимо. Что привело к тому, что применимость эффективных параллельных алгоритмов, предложенных в сообществе, занимающемся графовыми базами данных, до сих пор не изучена.

Тимофей. Важно, что для практического применения статического анализа необходимо не только обеспечить эффективное решение исходной задачи, но и предоставить пользователю (разработчику) механизмы дальнейшего анализа полученных результатов и их применения для решения прикладных задач. Одно из перспективных направлений — это создание интеллектуальных помощников на основе методов машинного обучения. Так, в работе Юлей Суй (Yulei Sui, Xiao Cheng, Guanqin Zhang, and Haoyu Wang, "Flow2Vec: value-flow-based precise code embedding 2020) предлагается подход, позволяющий использовать результаты поиска путей с контекстно-свободными ограничениями для улучшения качества работы прикладных решений, основанных на методах машинного обучения. Хотя в работе и рассматривается подход, основанный на линейной алгебре, не обсуждаются вопросы эффективной реализации соответствующих алгоритмов, не изучаются границы применимости данного подхода.

en

2.6 +Предлагаемые методы и подходы, общий план работы на весь срок выполнения проекта и ожидаемые результаты

ru

При разработке новых подходов и инструментальных средств для создания высокопроизводительных библиотек разреженной линейной алгебры для современного аппаратного обеспечения с использованием существующих языков программирования планируется использовать средства и методы метапрограммирования, в частности техники квазицитирования (code quotation) и генерации кода во время выполнения, что позволит бусшовно для конечного пользователя интегрировать высокоуровневый и низкоуровневый языки. В качестве низкоуровневого языка предполагается использовать OpenCL C, так как на сегодняшний момент это самый зрелый стандарт для разработки переносимого высокопроизводительного кода, выполняемого на различных устройствах, включая многоядерные процессоры, графические ускорители общего назначения, программируемые логические интегральные схемы (ПЛИС, FPGA). В качестве языка высокого уровня предлагается использовать язык F#, так как этот язык является функциональным языком программирования со строгой статической типизацией и развитыми срежствами метапроагрммирования. Кторме этого, это тязык интегрирован с платформой .NET, одной из самых популярных платформ для разработки прикладных решений. Вместе с этим, предполагается использование использование таких высокоуровневых средств организации асинхронных высчислений, как модель акторов, основанная на передаче сообщений между независимыми долгоживущими асинхронно выполняющимися процессами. Эти и другие абстракции представлены в языке F# и их использование должно позволить

построить средство разработки, упрощающее программирование асинхронных гетерогенных систем с несколькими графическими ускорителями.

При разработке нового программно-аппаратного стека, предназначенного для разработки высокопроизводительных решений на основе операций и примитивов разреженной линейной алгебры планируется использовать методы суперкомпиляции для оптимизации программ. В частности, планируется использование результатов Ильи Ключникова и его суперкомпилятора HOSC. Ожидается, что так как в нашем случае планируется создание библиотеки с ограниченным набором функций и типов, на ограниченном языке, проблему непредсказуемости результатов суперкомпиляции удастся решить, сузив общие решения до рассматриваемого частного случая. Вместе с этим, для обеспечения аппаратной поддержки планируется использовать подход, основанный на создании специализированной архитектуры по программе на функциональном языке программирования. В частности, планируется применять результаты, полученные Ричардом Таусендом. А именно, планируется на функциональном языке программирования реализовать базовый набор типов данных и операций, предусмотренных стандартом GraphBLAS, далее реализовать некоторые алгоритмы, используя реализованные операции, после чего оптимизировать их с применением методов суперкомпиляции, что, в частности, должно устранить создание промежуточных структур данных. Полученная программа будет транслироваться в специализированную аппаратную архитектуру, в результате чего будет получаться проблемно-специфичный процессор. Использование такого подхода вместе с ПЛИС, позволит получать специализированную аппаратную поддержку для конкретных задач и алгоритмов анализа данных.

При формальном описании свойств языка Cypher и доказательстве его свойств будет использоваться система автоматической проверки доказательств Coq, что позволит не только строго записать необходимые утверждения, но и гарантировать, что в них нет ошибок. На начальных этапах формализации планируется использовать результаты Надима Франсиса и Джозефа Мартона, содержащие описание семантики языка Cypher, однако, лишь на бумаге. В качестве первого шага планируется проанализировать эти описания и описать семантику в системе Coq. Далее планируется заняться описанием недавно предложенного расширения языка Cypher, позволяющего выражать регулярные и контекстно-свободные ограничения (Tobias Lindaaaker, Path Pattern, <https://github.com/thobe/openCypher/blob/rpq/cip/1.accepted/CIP2017-02-06-Path-Patterns.adoc>). После чего будет изучаться возможность трансляции конструкция языка Cypher (вместе с указанным расширением) в операции линейной алгебры. Задача здесь: описать максимальное подмножество языка, выразимое в терминах линейной алгебры. На этом шаге планируется опираться на результаты Флориса Гертса, касающиеся различных аспектов семантики языка MATLANG. Результаты будут зафиксированы в Coq и снабжены доказательствами корректности.

При изучении применимости алгоритмов, основанных на операциях линейной алгебры, к задачам статического анализа кода планируется использование результатов, полученных Рустамом Азимовым, Арсением Тереховым и Егором Орачевым. А именно, планируется использовать предложенные ими алгоритмы и, при необходимости, модифицировать с учётом особенностей решаемых задач. Кроме этого, планируется изучить применимость результатов работы таких алгоритмов для улучшения методов интеллектуального анализа кода с использованием методов машинного обучения. Здесь предполагается отталкиваться от результатов Юлей Суй и улучшить их.

июль 2021 – июнь 2022

Будут проведены разработка и экспериментальное исследование системы программирования графических процессоров на языке высокого уровня (F#). Будут выявлены основные возможности и ограничения прозрачной интеграции вычислений на графическом ускорителе общего назначения в язык высокого уровня со статической типизацией F# на основе методов метапрограммирования времени исполнения. Будет реализован прототип соответствующего инструмента, реализована библиотека базовых операций разреженной линейной алгебры, проведена её экспериментальное исследование, проанализированы результаты использования полученного средства разработки и самой библиотеки. На основе этого анализа будут сформулированы задачи на следующий год.

Будет выполнена формализация семантики ядра языка запросов Cypher с использованием системы автоматической проверки доказательств Coq. Начнутся подготовительные работы по формализации расширения, позволяющего выражать регулярные и контекстно-свободные ограничения. Начнётся обсуждение полученных результатов с сообществом GQL.

Будет проведено исследование применимости суперкомпиляции для оптимизации программ, написанных с использованием операций и примитивов линейной алгебры. А именно, будет проведён анализ существующих суперкомпиляторов для функциональных языков, выбран поддерживающий достаточно богатый для описания необходимых типов и функций язык. Далее, на выбранном языке будет реализована библиотека основных типов и операций разреженной линейной алгебры, после чего на основе этой библиотеки будут реализованы элементарные программы, эффект от специализации которых и будет изучаться. Планируется добиться устранения создания промежуточных матриц при выполнении последовательности арифметических операций над ними (сложение и умножение матриц), а также, при взятии маски. Для этого, вероятно, будут выполнены улучшения выбранного суперкомпилятора.

Будет проведено экспериментальное исследование алгоритмов поиска путей с контекстно-свободными ограничениями в контексте решения задач статического анализа кода на языке программирования Java. Для этого будет создана инфраструктура, включающая компоненты по построению необходимых графов и запросов (грамматик) по исходному коду (на основе инструментов типа Soot или WALA), выполнению запросов на посторонних графах с использованием алгоритмов, основанных на линейной алгебре, выполняемых на многоядерных ЦПУ и на графических ускорителях общего назначения, замеру требуемого для выполнения запроса времени и памяти. Результаты исследования будут подвергнуты анализу и сравнению с аналогичными результатами, полученными с использованием других алгоритмов. По итогам анализа будут сформулированы задачи по улучшению алгоритмов, которые будут решаться в следующем году.

июль 2022 – июнь 2023

По результатам предыдущего года будет вестись улучшение суперкомпилятора. Вместе с этим будет расширяться библиотека примитивов линейной алгебры функциями, предусмотренными стандартом GtaphBALS, что может потребовать дополнительных улучшений в суперкомпиляторе. Будет разработан набор базовых алгоритмов анализа графов на основе разрабатываемой библиотеки, таких как обход в ширину, построение транзитивного замыкания. Будет проведено исследование эффективности суперкомпиляции при оптимизации

таких алгоритмов. Будут выявлены и проанализированы случаи, когда полное устранение промежуточных матриц или векторов невозможно.

Начнутся эксперименты по трансляции выбранного функционального языка в специализированное аппаратное обеспечение. Будет проведён анализ влияния суперкомпиляции на производительность целевого решения. На данном этапе планируется сравнивать производительность с использованием симуляторов (типа modelsim). По итогам анализа будут сформулированы задачи по улучшению транслятора и суперкомпилятора, которые будут решаться в следующем году.

По результатам предыдущего года будет вестись улучшение средства программирования графических ускорителей общего назначения с использованием языка высокого уровня F#. Вместе с этим, будет вестись исследование применимости высокоуровневых абстракций асинхронного и параллельного программирования, предоставляемых языком программирования F#, для создания средств программирования гетерогенных систем и систем с несколькими графическими сопроцессорами общего назначения. Будет вестись разработка и реализация такого средства. Ожидается, что будет получен прототип, демонстрирующий возможности использования статических гарантий и создания обобщённых типов и операций при написании библиотек абстрактной (абстрагируются понятия моноида, полукольца) разреженной линейной алгебры.

Будет вестись формализация семантики расширения языка Cypher, позволяющего выражать регулярные и контекстно-свободные ограничения. Будет получено полное (с учётом указанного расширения) формальное описание неизменяющего графа подмножества языка запросов Cypher. Будут начаты работы по формальному описанию трансляции конструкций данного подмножества в операции линейной алгебры.

По результатам предыдущего года будет вестись улучшение алгоритмов поиска путей с контекстно-свободными ограничениями. Также начнётся изучение применимости результатов соответствующих анализов для улучшения результатов обработки программного кода методами машинного обучения.

июль 2023 – июнь 2024

Будут выполнены запланированные в предыдущем году улучшения суперкомпилятора и транслятора. Будет завершена адаптация инструмента компиляции функционального языка в специализированную аппаратную архитектуру к разработанному языку. Это позволит провести экспериментальное исследование полученного решения с использованием ПЛИС (FPGA). По результатам исследования и сравнения с аналогичными решениями, а так же с современными решениями, использующими многоядерные ЦПУ и графические ускорители, будут сделаны выводы о применимости выбранного подхода для создания программно-аппаратного стека для высокопроизводительных вычислений на основе операций линейной алгебры над разреженными структурами данных.

Будет завершена работа над прототипом средства программирования гетерогенных систем, содержащих графические ускорители общего назначения, с использованием языка высокого уровня F#. Будет проведено его экспериментальное исследование. Для этого с его помощью будет реализовано базовое подмножество стандарта GraphBLAS, реализованы некоторые алгоритмы анализа графов, использующие данное подмножество. Далее будет проведён

анализ как быстроедействие полученного решения по сравнению с решениями, основанными на других реализациях стандарта, так и особенности разработки с использованием разработанного средства. После чего будут сделаны выводы о применимости выбранного подхода к разработке высокопроизводительных абстрактных решений, использующих примитивы и операции линейной алгебры над разреженными структурами, с использованием языков высокого уровня.

Будет завершена работа по формальному описанию трансляции конструкций языка Cypher в операции линейной алгебры. Будет получена классификация конструкций на транслирующиеся в операции линейной алгебры и невыразимые в ней. На основе этого будут сформулированы рекомендации по разработке систем, использующих Cypher в качестве языка запросов, и матрицы и операции над ними для представления и обработки данных.

Будут созданы и проанализированы прикладные решения по статическому анализу кода, основанные на методах машинного обучения, использующих данные статического анализа, основанного на поиске путей с контекстно-свободными ограничениями. Будут сделаны выводы о применимости такого подхода для уточнения интеллектуальных средств анализа программного кода.

en

июль 2021 – июнь 2022

июль 2022 – июнь 2023

июль 2023 – июнь 2024

2.7 +Имеющийся у научного коллектива научный задел по проекту, наличие опыта совместной реализации проектов

ru

Руководитель проекта и многие его участники обладают опытом в разработке и исследовании алгоритмов, основанных на линейной алгебре, в том числе, алгоритмов поиска путей с контекстно-свободными ограничениями, являющегося одним из ключевых при межпроцедурном анализе кода, что подтверждается перечисленными ниже и некоторыми другими работами.

- Azimov, Grigorev, “Context-free path querying by matrix multiplication”, GRADES-NDA-2018.
- Orachev, Epelbaum, Azimov, Grigorev, “Context-Free Path Querying by Kronecker Product”, ADBIS-2020;
- Terekhov, Khoroshev, Azimov, Grigorev, “Context-Free Path Querying with Single-Path Semantics by Matrix Multiplication”, GRADE-NDA 2020.

Руководитель принимал успешное участие в совместной с Е.А. Вербицкой, А.В. Подкопаевым и Д.А. Березуном работе над проектам в рамках гранта РФФИ 18-01-00380. Также, С.В. Григорьев являлся исполнителем грантов РФФИ 15-01-05431 и Фонда содействия развитию малых форм предприятий в технической сфере (программа УМНИК, проекты N 162ГУ1/2013 и N 5609ГУ1/2014), руководителем гранта РФФИ 19-37-90101, а также является руководителем научной группы, в соавторстве с участниками которой опубликованы указанные выше и некоторые другие работы. Кроме этого, С.В. Григорьев являлся основным исполнителем гранта РФФИ 18-11-00100.

Р.Ш.Азимовым и С.В.Григорьевым предложен алгоритм поиска путей с контекстно-свободными ограничениями на основе матричных операций, доказана его корректность, получена оценка временной сложности (Rustam Azimov and Semyon Grigorev. 2018. Context-free path querying by matrix multiplication). Кроме того, предложено обобщение данного алгоритма, в котором в качестве ограничений над путями используются конъюнктивные грамматики, позволяющие выражать более сложные запросы к графам. Для обобщенного алгоритма также доказана корректность и получена оценка временной сложности. Также, совместно с другими участниками проекта, А.К. Тереховым и Е. Орачевым, предложены и другие обобщения этого алгоритма, в частности позволяющие найти один и все пути, удовлетворяющие ограничениям, или выполняющие поиск из заранее заданного множества вершин. Применимость именно этих алгоритмов к статическому анализу кода и предполагается исследовать, а в дальнейшем и модифицировать с учетом специфики возникающих в данной области задач.

А.К.Терехов, совместно с С.В.Григорьевым реализовал полнофункциональную поддержку запросов с регулярными и контекстно-свободными ограничениями для графовой базы данных (Terekhov, Grigorev, et al, "Multiple-Source Context-Free Path Querying in Terms of Linear Algebra принята на EDBT-2021). Для этого потребовалось реализовать расширение языка запросов Cypher, позволяющее выражать соответствующий тип ограничений и транслировать его в операции линейной алгебры. в терминах которых выражается используемый алгоритм поиска путей. Анализ выявленных в рамках данного проекта проблем послужил мотивацией для формализации семантики языка Cypher и построения его формального транслятора в операции линейной алгебры, а полученный опыт будет использован для решения поставленных уже в этом проекте задач.

Руководитель проекта принимал разработку системы Brahma.FSharp на основе языка программирования F#, позволяющую прозрачным для разработчика образом использовать в программах на F# код, написанный на языке OpenCL C, с сохранением гарантий иповой безопасности (Smirenko, Grigorev, "F# OpenCL Type Provider TeDe-2018). Этот проект будет взят за основу при разработке средств программирования графических процессоров с использованием языков высокого уровня.

Д.А. Березун имеет опыт исследований в области семантики языков программирования, метавычислений и программной специализации. В частности, им предложен алгоритм компиляции нетипизированного лямбда исчисления в низкоуровневое представление посредством игровой семантики программ и частичных вычислений (D.Berezun, N.D.Jones. Compiling untyped lambda calculus to lower-level code by game semantics and partial evaluation. 2017; D.Berezun, N.D.Jones. Working Notes: Compiling ULC to Lower-level Code by Game Semantics and Partial Evaluation. 2016). Кроме того, им была показана корректность предложенного

алгоритма и его обобщения, а также предложено обобщение понятия головной линейной редукции термов (Д.Березун. Полная головная линейная редукция. 2017).

Кроме этого, Д.А.Березуном, совместно с А.Тюриным и С.В.Григоревым получены результаты, показывающие применимость смешанных вычислений для оптимизации программ, выполняемых на графических процессорах общего назначения (Aleksey Tyurin, Daniil Berezun, and Semyon Grigorev, "Optimizing GPU programs by partial evaluation" PPop-2020). Результаты данного исследования послужат отправной точкой при исследовании применимости суперкомпиляции для оптимизации программ, построенных на основе операций линейной алгебры, так как они позволяют оценить эффект специализации, как частного случая суперкомпиляции, на выполнение программ на параллельных архитектурах.

Т.А.Брыксин

А.В.Подкопаев

А.!!Тюрин

Кроме этого, участники проекта создали набор данных, необходимый для экспериментального исследования разрабатываемых решений. Он представлен и используется в работе "Evaluation of the Context-Free Path Querying Algorithm Based on Matrix Multiplication". В ходе исследований планируется его использование и расширение при экспериментальных исследованиях различных алгоритмов.

2.8 +Перечень оборудования, материалов, информационных и других ресурсов, имеющихся у научного коллектива для выполнения проекта

ru

У коллектива имеется необходимое аппаратное и программное обеспечения для разработки и проведения экспериментального исследования алгоритмов, использующих многоядерные процессоры, графические ускорители общего назначения, а также для проведения иных запланированных задач.

2.9 +План работы на первый год выполнения проекта

ru

Планируется работа над заранее намеченными на этот год исследовательскими задачами, предоставление результатов на конференциях и подготовка результатов к печати. Также будет проведено осмысление полученных результатов с возможной формулировкой новых задач. Распределение задач между основными исполнителями проекта приведено в следующем разделе.

Также на первый год планируется 3 поездки с докладами на международные конференции (в среднем по 100000 рублей).

en

During the first year, it is planned to work on research questions listed in this plan, to present results at conferences, and prepare results for publication. Also it is planned to collaborate for understanding the new results. As a result, some new problems will be formulated. Detailed plan for each team member is presented below.

Also, 3 trips to international conferences are planned (on average, 100000 rub. each) in order to give talks.

2.10 +Планируемое на первый год содержание работы каждого основного исполнителя проекта (включая руководителя проекта)

ru

С.В.Григорьев, совместно с А.К.Тереховым займётся разработкой технологии высокоуровневого программирования массово-параллельных систем, позволяющей программировать графические процессоры общего назначения с использованием языка высокого уровня (F#) прозрачным для разработчика образом. Будут изучены возможности и ограничения трансляции подмножества языка F# в язык OpenCL C. Также будут изучаться основные принципы использования подобного инструмента с целью поиска баланса между высокоуровневыми абстракциями и необходимостью детального контроля для достижения высокой производительности. Также будет проведено экспериментальное исследование полученной реализации на примере разработки библиотеки базовых операций линейной алгебры: умножения разреженных матриц, поэлементного сложения разреженных матриц.

Т.А.Брыксин, Р.Ш. Азимов и Е.Орачев займутся экспериментальным исследованием алгоритмов поиска путей с контекстно-свободными ограничениями, основанных на операциях линейной алгебры, применительно к статическому анализу кода, в том числе статистическими методами, в частности методами машинного обучения. Будет проведено экспериментальное исследование алгоритмов поиска путей с контекстно-свободными ограничениями для решения задач межпроцедурного потоко- и контекстно-чувствительного анализа указателей для языка программирования Java. Также будет проведена оценка пригодности результатов данных анализов для дальнейшей статистической обработки и применимость для решения прикладных задач анализа кода.

Формализацией семантики языка зарпосов Cypher с использованием системы автоматической проверки доказательств Coq займётся А.В.Подкопаев совместно с С.В.Григорьевым. На первом этапе планируется формализации ядра языка Cypher, а также описание его семантики в терминах реляционной алгебры в системе Coq.

А.Тюрин и Д.А.Березун будут заниматься вопросами применимости суперкомпиляции для оптимизации алгоритмов, реализованных на основе операций линейной алгебры. В частности, будет проведён анализ существующих решений в области суперкомпиляции с целью выбрать наиболее подходящий для исследуемых задач язык и суперкомпилятор для него. Ожидается, что будет найден модельный функциональный язык, поддерживающий необходимые конструкции и суперкомпилятор для него. Далее будет выполнена реализация базовых

типов данных, таких как разреженная матрица в виде дерева квадрантов или списка координат ненулевых элементов, и основных операций (поэлементное сложение матриц, умножение матриц, применение маски) на выбранном подмножестве функционального языка. После чего будет проведено экспериментальное исследование суперкомпилятора на простых программах, использующих реализованные типы данных и операции. Например, последовательное сложение нескольких матриц или применение маски к результату некоторой операции. Будет проведён анализ того, какие структуры данных и принципы написания операций (кода) позволяют достичь наилучшего результата в смысле производительности итогового решения.

К обсуждению всех задач, работе над ними, и написанию статей будут привлекаться включённые в состав научного коллектива студенты, магистры и аспиранты.

2.11 +Ожидаемые в конце первого года конкретные научные результаты

ru

Будет разработан подход на основе метапрограммирования для программирования графических процессоров с использованием языков высокого уровня. Будет реализован прототип соответствующего инструментального средства и проведено его экспериментальное исследование. По итогам, одна работа будет представлена на конференции. Результаты будут опубликованы в сборнике докладов, индексируемом в Scopus.

Будет проведено экспериментальное исследование применимости суперкомпиляции для оптимизации алгоритмов, выраженных в терминах операций линейной алгебры. Полученные результаты будут проанализированы, на основе результатов анализа будут сформулированы конкретные задачи, решение которых планируется на второй год работы.

Будет формально описана семантика ядра языка запросов Cypher (неизменяющее граф подмножество) с использованием инструмента Coq. Результаты будут представлены сообществу GQL, представлены на профильной конференции и опубликованы в сборнике материалов индексируемом в Scopus.

Будет проведено экспериментальное исследование применимости алгоритмов поиска путей с контекстно-свободными ограничениями, основанными на операциях линейной алгебры, в рамках задач статического анализа кода. Будут проанализированы полученные результаты и сформулированы направления и конкретные задачи по улучшению рассмотренных алгоритмов. По итогам, одна работа будет представлена на конференции. Результаты будут опубликованы в сборнике докладов, индексируемом в Scopus.

Над прочими заявленными темами будет вестись работа, однако результаты будут опубликованы во второй год проекта.

en

Results will be presented at a conference and published in proceedings indexed in Scopus.

The work on other topics will have a progress, but the results will be published during the second year of the project.

2.12 +Перечень планируемых к приобретению руководителем проекта за счет гранта Фонда оборудования, материалов, информационных и других ресурсов для выполнения проекта

ru

Не более 800 тыс. рублей ежегодно будет тратиться на поездки с докладами на конференции.
Расходов на оборудование не предполагается.